Prenons la main de la mémoire
Mes premiers pas en informatique (1964-1982)

Pierre Lescanne

Les mathématiciens méprisent, avec beaucoup d'aristocratisme, les calculateurs.

François le Lionnais Un Certain disparate. Entretien avec Jean-Marc Lévy-Leblond et Jean-Baptiste Grasset, (1976) in Oulipo. Chap. 37.

Cet article présente mon initiation à l'informatique durant la période 1964-1982 et je vais tenter de retracer en quoi consistait cette initiation, sachant que comme tous les gens de ma génération, je suis essentiellement autodidacte ; en effet, l'enseignement de l'informatique que nous recevons avant et après 1970 est minimum. Nous prenons quelques cours certes, mais l'essentiel de notre apprentissage se fait, par la lecture de livres ou d'articles, peu nombreux, parfois par la redécouverte de concepts, par quelques rares conférences et par les écoles de printemps et d'été, qui se mettent en place. Notre méconnaissance de l'anglais est un énorme handicap. En effet, dans ma formation du collège et du lycée, je n'ai appris que l'allemand, le latin et le grec, ce qui n'est d'aucune aide, compte tenu du fait que, dès cette époque, même les Allemands et les Suisses publient en anglais. De plus, immédiatement acquis, les concepts et les contenus scientifiques doivent être enseignés aux autres dans cette construction ininterrompue d'une science nouvelle à laquelle nous participons.

La table 1 retrace les principales étapes de l'émergence de l'informatique à Nancy, ainsi que ma propre carrière qui s'y insère. Cependant, j'adopterai ici dans ma démarche une approche parfois un peu technique et globalement subjective. En effet, je n'analyserai pas globalement l'apparition de l'informatique dans le milieu universitaire nancéien, mais je dirai comment j'ai vécu ces années d'apprentissage et de construction de la science qu'est maintenant l'informatique. Je présenterai comment cette activité s'est, chez moi, articulée avec un enseignement de mathématiques, compte-tenu de ma formation initiale et de la rareté des postes en informatique à cette époque et compte-tenu aussi de mon détachement au CNRS [1] .

Mon but est de faire connaître aux générations plus jeunes, qui ont toujours baigné dans l'informatique, ce que nous, les pionniers, avons vécu. Mais comme je ne considère que le petit bout de la lorgnette, j'invite donc le lecteur qui souhaite un peu plus d'objectivité sur les débuts de l'informatique à Nancy à écouter la conférence enregistrée de Marion Créhange [6] ou à lire, les écrits de Claude Pair [44], de Marion Créhange et Marie-Christine Haton [7], de Pierre-Éric Mounier-Kuhn [42] ainsi que ceux dont je suis l'auteur ou le co-auteur [37, 8]. De plus une table donne en annexe C, une description succincte des langages de programmation que j'ai appris à utiliser, tandis que l'annexe A est un glossaire des termes techniques que j'utilise. Enfin, il est fortement recommandé de lire ce document sur une tablette ou un ordinateur connecté pour bénéficier des hyperliens, car nous sommes au XXI$^{\text{ème}}$ siècle !

1 Mes études supérieures

Je suis né à Dakar le 22 mars 1947 et j'ai fait mes études secondaires à Colmar. Au moment des études supérieures, à l'automne 1964, je choisis Nancy, parce que ma grand-mère y habite et que je peux loger chez elle, mais aussi parce que je sais qu'en mathématiques c'est un endroit où

1. Je quitterai le CNRS en 1997 pour devenir professeur à l'École normale supérieure de Lyon.

1957	Installation d'une IBM 604, machine à programmes câblés
1958	Installation d'un IBM 650, premier ordinateur
1958	Création du cours d'*Analyse et calcul numérique*
1959	Création du centre de calcul
1964	Mon entrée à la Faculté des Sciences de Nancy
1965	Installation d'une CAE 510
1967	Création du département informatique de l'IUT
1968	Ma nomination comme assistant délégué (non titulaire) – mon premier cours de programmation
1970	Installation du CII-10070
1971	Ma thèse de 3ème cycle
1971-72	Mon service militaire
1972	Ma nomination comme maître assistant à l'Université de Nancy II
1973	Création de l'équipe de recherche associée au CNRS
1974	Mon détachement au CNRS
1976	Création du CRIN, laboratoire associé au CNRS
1979	Ma thèse d'État
1980	Mon départ au MIT

TABLE 1 – Les étapes de l'informatique universitaire à Nancy et de ma carrière

s'y font de bonnes choses. Ayant lu en terminal, le livre édité par François Le Lionnais *Les grands courants de la pensée mathématique* [26], je sais que Nancy a une réputation d'excellence héritée du temps où elle hébergeait les bourbakistes et je connais un peu les enjeux de la recherche en ce domaine, sans toutefois, bien sûr, les maîtriser. Quand les cours de Claude Georges, notre professeur en première année, nous conduisent à aborder les *familles sommables* (les séries [2] étant considérées par lui comme un peu trop contraintes) dès le mois de décembre et un peu plus tard au mois de janvier l'intégrale de Stieltjes sur les fonctions à variation bornée (là encore parce que l'intégrale de Riemann sur les fonctions continues n'est pas assez générale), je suis moins submergé que mes condisciples, pour lesquels cette « pédagogie » de choc fait l'effet d'une douche glacée [3]. Nous étions dans l'enseignement supérieur depuis un peu plus de deux mois et n'avions jamais entendu parler au lycée de séries et d'intégrales. En 1967, je me suis retrouvé « licencié », la licence étant le diplôme qui permet de se présenter aux concours de l'enseignement et au DEA [4]. Mais je sais que je veux faire de la recherche et un DEA de math me paraît la chose la plus normale. Je m'y inscris par conséquent. Mais j'ai plusieurs activités annexes : je fais de la voile, je joue au rugby (avec mes 78 kg, comme je suis grand, je suis deuxième ligne !), je suis investi à l'UNEF (Union Nationale de Étudiants de

2. En mathématiques, les *séries* sont les sommes infinies qui peuvent diverger ou converger. Dans les *familles sommables*, il n'y a pas d'ordre sur les termes. Si l'on compare l'enseignement de *Mathématique générale et physique*, ou MGP, que l'on appelle propédeutique et qui est la première année de faculté des sciences, avec celui des classes préparatoires aux grandes écoles, on note que les séries n'étaient pas au programme de la première année de math-sup, mais seulement à celui de la deuxième année de math-spé, tandis que les familles sommables n'étaient pas évoquées du tout.

3. Je découvre plus tard que Claude George fonde son cours d'analyse de MGP, sur le livre de Walter Rudin *Principles of Mathematical Analysis* ; ce livre est destiné aux étudiants de quatrième année de bachelor ou de première de master des universités américaines. Le cours de Math 1 de Pierre Eymard (deuxième année de fac pour nous) est fondé sur le livre de Walter Rudin *Real and Complex Analysis* que nous nous procurons d'ailleurs dans un achat groupé sur la recommandation de notre professeur. Ce livre est très clairement, dans l'esprit de son auteur, un cours de master (Bac+5). C'est d'ailleurs, dans ce livre que je me suis initié à l'anglais.

4. Le DEA est le Diplôme d'Études Approfondies. Il correspond grosso modo au master d'aujourd'hui après quatre ans d'études.

France)[5]. Je me souviens, en particulier, de soirées et de nuits à l'AGEN[6] à écouter les joutes oratoires entre les « meuleus »[7] et les communistes de l'UEC[8], de stricte obédience ; nous, les cathos de gauche[9], comptons les points. Au cours de l'une d'elles, au mois de mars 1968, nous allons à une heure du matin soutenir les étudiants de la cité universitaire qui demandent pour les étudiants des deux sexes l'autorisation de circuler librement dans les cités universitaires y compris dans les cités de filles et qui se font molester par les CRS sur le Cours Léopold[10]. Mai 68 couve et le printemps étudiant éclate à Nancy ce jour-là. Je suis au front à la Faculté des sciences de Nancy, où rétrospectivement on peut dire que ça été relativement calme, quoique que j'apprendrai ultérieurement que deux éminents professeurs en ont été très affectés. Nous, les étudiants syndicalistes « reconstruisons » l'Université et rencontrons les professeurs syndiqués, dont l'un d'eux est au SGEN[11] et avec lequel, je sympathise ; c'est Jean-Louis Ovaert [5].

Rebelle[12], je suis dans mon engagement au syndicat étudiant, et rebelle je suis en mathématiques. De jeunes assistants montent, dès 1967, un séminaire de recherche alternatif, dans une annexe du département de mathématiques, rue Sellier à Nancy[13] et nous y sommes conviés ; nous nous y rendons, Jean-Luc Rémy[14] et moi, comme des conspirateurs. Quand il s'agit de choisir mon sujet de mémoire de DEA, Pierre Eymard qui dirige le DEA nous propose de lire un article de mathématiques, parmi une liste qu'il nous soumet, pour en faire ensuite un compte-rendu écrit. Comme au séminaire alternatif, j'ai entendu parler des « mesures idempotentes » je choisis l'article de P. J. Cohen[15] de 1960 *On a conjecture of Littlewood and idempotent measures* [4]. Mes collègues du séminaire alternatif me conseillent en fait de lire et de rapporter sur un autre article intitulé *A simple proof of the theorem of P. J. Cohen*[16] [21]. Cela plaît à Pierre Eymard, spécialiste d'analyse harmonique, thème de l'article et donc au jury, puisque j'obtiens mon DEA sans problème en septembre 1968.

Suite au décès d'une enseignante chercheuse, dans un accident de voiture[17], un poste d'as-

5. Je suis donc à gauche, mais pas à l'extrême gauche. Un jour de mars 1968, je dois représenter les étudiants en science à une conférence sur l'avenir de l'Université européenne au Centre des Prémontrés de Pont-à-Mousson. J'y rencontre un professeur tchécoslovaque tout heureux de se trouver à l'Est ; nous sommes en plein printemps de Prague et il a pu répondre à l'invitation des organisateurs ce qui le réjouit ; il m'offre un paquet de cigarettes pour mon anniversaire en ce deuxième jour du printemps, le 22 mars 1968. Ce même jour, au cours d'un exposé assez ennuyeux, un orateur parle de la relativité d'Einstein. Dans mes études, on ne me l'a jamais enseigné, je demande donc à mon voisin physicien qui représente le SNESUP (syndicat national de l'enseignement supérieur), dont il est le secrétaire nationale, de me l'expliquer. C'est ainsi que le 22 mars 1968 Alain Geismar m'initie à la théorie de la relativité.

6. Association générale des étudiants de Nancy.

7. Les « meuleus » sont les membres de l'Union des jeunesses communistes marxistes-léninistes. Meuleu vient du fait qu'ils sont ML (marxistes-léninistes), en fait prochinois, c'est-à-dire alignés sur Pékin.

8. Union des étudiants communistes, alignée sur Moscou.

9. Qui ne sommes pas alignés sur Rome ! Du moins, nous le prétendons !

10. Place centrale de Nancy, lieu de vie étudiante.

11. Syndicat général de l'Éducation nationale, CFDT (Confédération française démocratique du travail).

12. Ce qualificatif m'a été donné, des années après, par Françoise Bellegarde, pour qui, je suis scientifiquement et universitairement rebelle. Françoise Bellegarde (1942-2016), qui a d'abord été professeure de mathématique en lycée, fera, avec Claude Pair, une thèse de 3ème cycle sur la compilation, puis, avec moi, une thèse d'État sur la programmation fonctionnelle. Elle séjournera plusieurs années aux États-Unis, puis sera professeure à Besançon, jusqu'à sa retraite.

13. Le département de mathématiques est situé, rue de la Craffe, au cœur de la faculté des sciences où nous avons nos cours. Le bâtiment est aujourd'hui occupé par un collège.

14. Jean-Luc Rémy est le plus brillant étudiant en mathématique de ma génération, dans mon environnement nancéien. Il fera un carrière de chercheur en informatique au CNRS et sera connu pour son algorithme de génération aléatoire d'arbres binaires [54].

15. Médaille Fields, en 1966, mais je ne fais pas le lien.

16. On notera le terme « the theorem », comme s'il n'y a aucune ambiguïté sur le théorème de P. J. Cohen dont il s'agit.

17. Joëlle Rousseau.

sistant se libère. Claude Pair, que je ne connais pas encore à l'époque, en parle à son grand ami Jean-Louis Ovaert et je me retrouve embauché, pour enseigner les maths. Ce sera en PC1 (première année du cycle Physique-Chimie). À l'époque, les recrutements se font à la bonne franquette. Je fais un choix, pour l'année 1968-1969 : je commence à enseigner tout en préparant l'agrégation de maths, si ça passe, tant mieux, sinon, je commencerai une thèse de troisième cycle dans une discipline à définir entre maths et la toute nouvelle informatique. Ovaert m'a dit du bien de son ami Claude Pair et les mathématiques pures, qui m'avaient beaucoup plu, en tant que matière d'enseignement, me tentent moins pour y faire de la recherche, car elles me paraissent ne traiter que de sujets avec un moindre intérêt social, comme c'est le cas des « mesures idempotentes ». On ne peut pas raconter à son frère ou à sa sœur ce qu'on fait et parfois même pas à un autre mathématicien. D'autre part, la recherche en mathématiques pures, malgré la dynamique du séminaire alternatif ne semble pas être faite en équipe et, si équipes il y a, elles sont à Paris. La présence d'un vrai leader qui s'implique, d'une ambiance, d'un domaine plein de promesses me titille. Quand l'étape de l'agrégation est franchie avec succès, mon choix est fait, je ferai de l'informatique [18], mais je garderai toujours de bons contacts avec les mathématiciens et avec les mathématiques. Concernant l'agrégation, mon succès à ma première tentative et mon classement me surprennent, car cela se passe dans une petite préparation de province. À l'écrit de l'agrégation de maths, il y a deux épreuves de six heures et une de quatre heures. L'équipe de rugby de la Faculté des Sciences à la création ex nihilo de laquelle j'ai participé, est dans la phase finale du championnat d'académie. Pour cause d'agrégation, je ne peux pas participer à un match qualificatif, donc important, mais le capitaine insiste pour que je vienne. Je réponds que ce jour-là j'ai la dernière épreuve de l'agrégation qui se termine à 14h, or le match commence à 13h30. Le capitaine me dit « Mais alors, tu peux venir pour la deuxième mi-temps ». Donc, dès ma copie rendue, j'enfourche mon vélo, je me rends au stade, je me change et je rentre sur le terrain à ma place de numéro 4. Dans les cinq minutes qui suivent, mon énergie accumulée par 16 heures d'épreuves intenses de maths, se libère sur un pauvre adversaire que j'assomme d'une manchette. Aujourd'hui ça vaudrait assurément un carton rouge, mais à l'époque, ça ne se fait pas, et nous écopons d'un simple coup franc. Je me calme cependant par la suite et je crois me souvenir que nous avons gagné. L'année d'après, nous avons été champions d'académie, ce qui fait rire mon fils ancien étudiant à Toulouse.

Je suis assistant à la Faculté des sciences jusqu'en 71, année de mon service militaire. Claude Pair enseigne le cours d'algèbre et je suis une année son assistant pour les travaux dirigés. En 1970, avec Robert Mainard [19], mon professeur de physique de MGP, qui en est à l'origine, je participe à la mise sur pied d'une filière de premier cycle *Mathématique, Physique et Technologies*, qui se veut être une alternative aux autres filières scientifiques, à savoir MP (Mathématiques et Physique) et PC (Physique et Chimie) par son contenu et son orientation résolument technologique. Je suis chargé d'un enseignement intégré (cours et TD) de mathématiques devant une promotion d'une vingtaine d'étudiants. J'ai 23 ans, je ne suis guère plus âgé qu'eux.

2 Mes premiers cours d'informatique

Si je veux un jour faire de l'informatique, il faut que j'apprenne la programmation ! Or, à l'automne 1968, un cours d'Algol 60 est proposé et est enseigné par Christiane Legras [20]. Je me

18. Nous employions le terme depuis au moins 1967, date de l'inauguration du département informatique de l'IUT qui doit sa création au Ministre Christian Fouché, qui avait fait l'annonce d'un département de « carrière de l'informati**que** », au lieu de « carrière de l'informati**on** ». Les deux départements seront alors créés.

19. Robert Mainard fut ensuite le troisième président de l'Université scientifique de Nancy.

20. Fille de Jean Legras [27], fondateur du centre de calcul de Nancy et initiateur du calcul numérique à Nancy. Jean Legras était à l'origine mécanicien avec une thèse sur l'aile portante, mais il deviendra un spécialiste

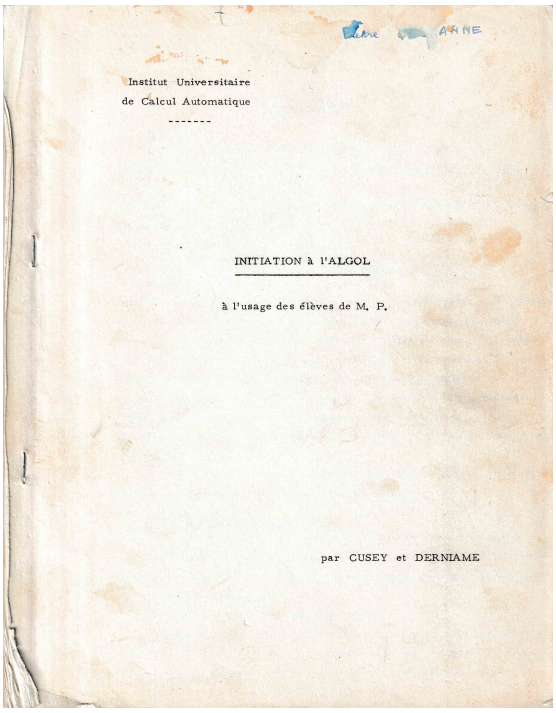


FIGURE 1 – Mon poly d'Algol 60.

souviens de ce cours au tableau noir dans une salle du bâtiment du Centre de calcul, en face du *Trésor de la langue française*, avenue de la Libération, près de la place Godefroy de Bouillon. La bâtiment est moderne, différent de ceux de l'Institut de mathématiques et de physique de la Porte de la Craffe. Nous avons eu comme support de cours le polycopié *Initiation à l'Algol à l'usage des élèves de M. P.* par Cusey et Derniame [9] (figure 1 et le plan du cours en Annexe B). Au milieu des mots-clés [21] **début**, **fin**, **si alors sinon** et des **;** de cette version francisée d'Algol 60, Christiane nous propose un exemple que je ne comprends pas ; en effet, je n'ai pas fait d'analyse numérique, ni a fortiori de programmation, ni encore moins d'algorithmique et je ne reconnais pas, dans l'instruction `x := (x / 2) + (1 / x);` la suite $x_{n+1} = \frac{x_n}{2} + \frac{1}{x_n}$ qui calcule $\sqrt{2}$. Pour quelqu'un qui, comme moi, a appris une certaine dose de maths, c'est vexant ! Mais ça me fait comprendre sur quoi, en pédagogie, peut reposer un blocage. Ça me fait aussi comprendre que l'enseignement de mathématiques pures que j'ai suivi est particulièrement abstrait. Mais je me souviendrai de cette suite, car je la prendrai, comme exemple, neuf mois plus tard, pour introduire ma leçon d'agrégation intitulé *suite de Cauchy* ; en effet, cette suite est effectivement un bel exemple de suite de rationnels qui ne converge pas vers un rationnel. Pour calculer cette suite il faut utiliser une boucle POUR dont la sémantique (ce qu'elle signifie) nous est expliqué dans l'organigramme de la figure 2. Étant donné qu'aujourd'hui, j'éprouve des difficultés à comprendre cet organigramme. je me demande ce qu'il en était en 1969. En fait, dans ce cours d'Algol-60, nous apprenons plus la syntaxe que la sémantique. Mais nous apprenons

d'analyse numérique, qu'il enseigne à la faculté des sciences et dans les écoles d'ingénieur. Il est l'auteur de plusieurs livres d'enseignement de l'analyse numérique [28, 29].

21. Voir le glossaire en fin d'article.

surtout comment écrire un programme qui sera accepté par le compilateur. Nous savons pas encore qu'affirmer qu'un programme est correct, ce n'est pas cela. Nous commencerons à le comprendre quand nous élaborerons la « théorie des programmes » (partie 2 de l'article). De plus, maîtriser Algol 60, c'est maîtriser les entrées-sorties, comme cela est décrit dans le chapitre IV du polycopié (annexe B). C'est indispensable, mais un peu rébarbatif et cela n'a pas l'élégance des autres constructions. Leur nécessité s'impose si l'on veut écrire des programmes qui produisent des résultats qui sortiront sur une imprimante, car c'est la seule sortie possible, lisible par le commun des mortels. Les autres médias, à savoir les rubans perforés et les cartes perforées ne sont pas d'une grand utilité pour tester si notre exercie a été concluant. Comme le dit le polycopié, ces instructions sont spécifiques à la CAE 510 [22]. À part cela, Algol 60 me plaît beaucoup et l'algorithmique qu'il sous-tend aussi. Avec ce cours, il n'y a pas de passage sur machine. Plus tard, j'essaie mon premier programme Algol 60 (très probablement un calcul de factoriel) sur la CAE510 du centre de calcul avec sortie sur son imprimante, qui est une machine à écrire IBM à boule (figure 7). Auparavant, il aura fallu charger le compilateur qui est sur un ruban de papier perforé. Tout un art ! Il ne faut surtout pas le déchirer ! Quelques temps après, je suis un cours de COBOL, mais je comprends rapidement que ce langage n'est pas pour moi. Je n'en retiens que ses quatre divisions et ses PIC 999, car, si j'ai bien compris, l'important dans ce langage, ce sont les formats.

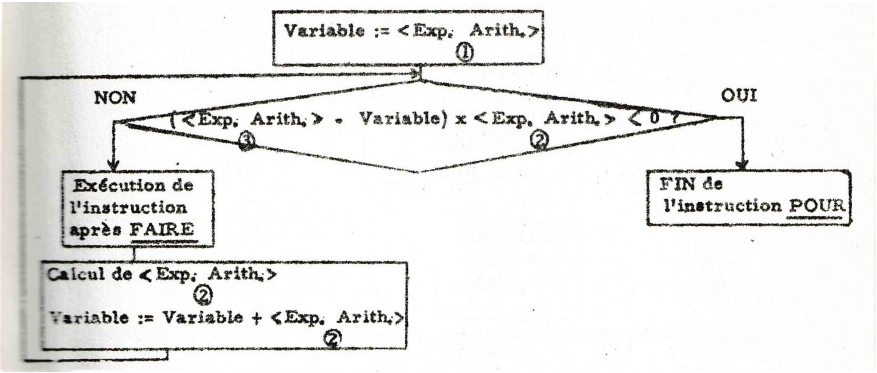


FIGURE 2 – L'explication de la boucle POUR par un organigramme, dans le polycopié.

Mon ami Jacques Guyard [23] m'a raconté une anecdote qui illustre bien le monde informatique d'alors. Fasciné par un cours d'Algol 60 (peut-être le même que le mien) où on avait dû lui exposer un autre grand classique du genre, à savoir le triangle de Pascal, il veut passer à la pratique. Or il vit dans un monde de physiciens où le langage de référence est FORTRAN. Son laboratoire possède des programmes qui calculent des nombres similaires utilisés par les physiciens. Ces programmes et leurs sous-programmes ne sont pas récursifs et Jacques Guyard s'est dit qu'en les rendant récursifs il en simplifierait le code. C'est ce qu'il fait. Le compilateur ne réagit pas à ces appels imbriqués de sous-programme et produit un code exécutable. Mais bien

22. La CAE 510 est une machine qui n'a pas de système d'exploitation. Elle est donc mono-utilisateur et nécessite pas mal de manipulations pour la faire fonctionner comme l'on veut. Les concepteurs du compilateur ALGOL 60 n'ont pas pu s'appuyer sur un système d'exploitation pour leurs entrées et sorties et donc ils ont dû tout concevoir à partir de zéro. De plus, le langage machine est spécifique. Ils n'ont pas pu envisager une réutilisation de leur travail.

23. Par la suite, Jacques Guyard est devenu enseignant-chercheur en informatique, professeur et directeur de l'École d'ingénieur ESIAL, qui deviendra Télécom Nancy.

sûr, ce code fait n'importe quoi. En se renseignant, il comprend que FORTRAN à la différence d'Algol 60 n'autorise pas les appels imbriqués de sous-programme, ce qui est le principe de la récursivité et que le compilateur FORTRAN ne prend pas la peine d'en avertir l'utilisateur, tant la récursivité parait incongrue, mais il « compile » néanmoins le programme. En FORTRAN, un appel à un sous-programme n'est rien de plus qu'un branchement [24].

Pour compléter ma formation, je prends un « vrai » cours de programmation : le certificat C1 de la maîtrise d'informatique qui vient d'être créée. J'y apprends les fondements de la programmation de l'époque à savoir les registres, l'adressage indirect, le calcul d'instruction, les instructions de base comme LOAD ou les branchements, bref le langage machine ainsi que l'assembleur et son langage. J'y apprends aussi ce qu'est un système d'exploitation ou comment une machine boote. Le professeur est Jean-Claude Derniame, maître-assistant [25] à l'époque et son assistant pour les travaux dirigés est Jacques Ducloy [26], qui a étudié à l'ENSEM quand j'étais à la faculté. Il a suivi un cours similaire, l'année précédente, ce qui l'autorise à être notre enseignant. D'une année sur l'autre le contenu du cours évolue au grès de l'acquisition des connaissances et des progrès de la technologie. Donc on ne peut même pas dire que Jacques ait suivi le même cours l'année précédente, mais il prend sa fonction très à cœur. Ainsi comme le centre de calcul a acquis un nouvel ordinateur : une CII 10070, nous apprendrons son langage d'assembleur, le SYMBOL, et son métalangage d'assembleur, le METASYMBOL. La documentation de ces langages, surtout du METASYMBOL est insuffisante et probablement incohérente, si bien que nous devrons nous habituer à voir la doxa changer d'une semaine sur l'autre. Nous coopérons avec notre enseignant et ami pour essayer de dégager les concepts de ce langage qui mélange, dans la même syntaxe approximative, le langage d'assembleur et ce que l'on peut dire sur lui. En effet, puisqu'en assembleur, on répète plusieurs fois les mêmes instructions à des nuances près, une suite d'instructions en METASYMBOL, n'est pas un programme, à proprement parler, mais une description visant à produire, puis à faire interpréter un programme d'assembleur en SYMBOL. On s'y perd un peu, entre tel mot clé qui appartient au langage d'assembleur et tel autre qui appartient à son métalangage. Qu'est-ce que dit le manuel ? Le dit-il vraiment ? D'ailleurs nous n'avons que des photocopies d'extraits parmi ceux qui sont pertinents. Et puis, ce cours n'est que théorique, puisque nous n'avons pas encore la machine.

Nous apprenons aussi les *cartes de commande* du système d'exploitation, qui sont des consignes à l'ordinateurs. On parle de « cartes » puisque a priori nous entrons nos programmes par cartes perforées. Même quand nous rentrerons les programmes à la console, la syntaxe bizarre par carte (ou ligne) de commande sera conservée. La figure 3 est issue d'un listing relatant, plusieurs année plus tard, l'exécution du programme Pascal de calcul symbolique dont je parle plus loin. Puisque le système d'exploitation fonctionne en *traitement par lot* [27], il faut lui dire comment prendre en compte mon travail (mon job) quand ce sera le moment, pour lui, de le faire. La syntaxe des « cartes » est incohérente et incompréhensible et n'a absolument aucune logique. La seule cohérence entre toutes les cartes est le fait qu'elles commencent par

24. Le grand spécialiste de la récursivité, Henry Gordon Rice (le célèbre théorème de Rice est une de ses contributions majeures), écrit, en 1965, un article [47] où il montre, ce dont on n'est pas certain à l'époque, à savoir qu'en FORTRAN on peut coder n'importe quelle fonction récursive au sens mathématique du terme, au prix du codage de la pile de récursion par des tableaux. Il implante pour cela la célèbre fonction d'Ackermann, dont il donne un programme FORTRAN d'une trentaine de lignes et un organigramme pour en décrire la structure, alors que le code Algol 60 de la même fonction tient en quelques lignes et n'a pas besoin d'explications supplémentaires.

25. Il devenu professeur à l'Université de Lorraine.

26. Jacques Ducloy a fait l'*École nationale supérieure d'électricité et de mécanique* (ENSEM). Il deviendra ingénieur de recherche au CNRS, d'abord à l'IUCA (Institut universitaire de calcul automatique), puis à l'INIST (Institut de l'information scientifique et technique), puis au LORIA, qui est le laboratoire de recherche en informatique de Nancy.

27. Je découvrirai les systèmes en temps partagé au MIT en 1980.

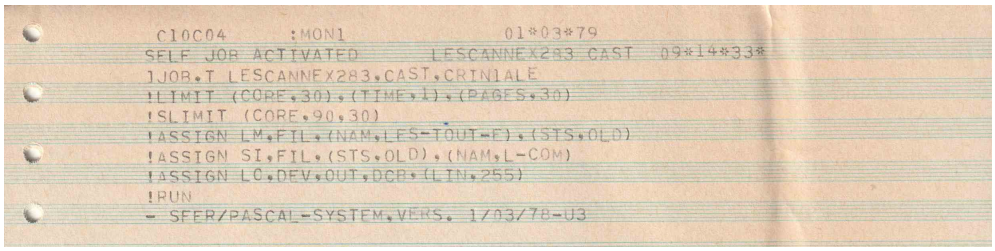


FIGURE 3 – Listing avec ses « cartes de commandes »

un point d'exclamation « ! », suivi du nom de la commande. Il y a les cartes LIMIT et SLIMIT qui énoncent les limites en temps, en mémoire interne et en mémoire externe dans lequel mon programme a le droit de s'exécuter et les tailles des sorties qu'il a le droit d'imprimer [28]. Si je choisis des limites trop grandes, mon programme sera rejeté du lot. Il y a la carte RUN qui demande l'exécution d'un programme donné, ici le sous-système SFER/PASCAL-SYSTEM qui sert d'environnement à mon programme. La pire des cartes de commande est la carte ASSIGN qui indique où lire les données, soit sur un disque fixe, soit sur un disque amovible, soit sur un lecteur de bande magnétique, soit sur une lecteur de cartes perforées et où produire les sorties sur les mêmes, auxquels il faut ajouter une imprimante à tambour. Chaque support physique (device) a sa propre syntaxe, puisque, avec le recul, j'ai l'impression qu'elle a été conçue sur un canevas commun par des programmeurs différents qui l'ont défini au moment d'écrire cette partie de logiciel. Mon listing de la figure 3 comporte trois cartes ASSIGN. Si ma mémoire est bonne, l'une dit où le système doit trouver mon programme à exécuter, c'est-à-dire dans le fichier LES-TOUT-E, où il doit trouver les données de mon programme, où il doit écrire les résultats de mon programme, en l'occurrence sur l'imprimante à tambour et que la dite imprimante utilise du DCB, c'est-à-dire du décimal-codé-binaire [29]. La figure 3 illustre bien le type de sortie produit par une telle imprimante. Quand le programme est rejeté du lot à cause d'une carte de commande erronée, c'est l'humiliation, car il faudra attendre le prochain tour (le prochain lot), qui peut être dans un quart d'heure ou une demi-heure ou le lendemain. Nous sommes très conservateurs et nous nous contentons de recopier les exemples qu'on nous a donnés sur des pages manuscrites photocopiés, en n'essayant de ne faire aucune erreur dans les parenthèses, les virgules et les mots clés et en essayant d'identifier les parties qui changent pour notre programme et celles qui ne sont pas susceptibles de changement. Il est clair que cette pédagogie n'est pas basée sur des principes et des concepts, mais sur des exemples à imiter, un peu comme les scribes babyloniens apprenaient l'algorithmique [22].

Pour l'initiation pratique, nous avons, comme pour toute formation technologique, ce que nous appellerions aujourd'hui un *projet* semestriel, en l'occurrence un *projet logiciel* [30] puisqu'il s'agit d'une formation à l'informatique. Nous devons écrire un assembleur (un programme de traduction) d'un langage d'assembleur ad hoc vers une machine virtuelle [31] et donc simulée dite

28. Il n'est pas rare malgré ces contraintes de voir l'imprimante échapper à son dresseur et produire en musique des pages de 0 et de 1. Cette musique anticipe celle des *Tambours du Bronx*. Ça fera du papier pour les classes de maternelles !

29. On notera que les reste de la carte utilise l'anglais, mais que l'abréviation DCB est du français, puisque « décimal codé binaire » se dit en anglais *binary coded decimal* et s'abrège BCD.

30. Le mot « *logiciel* » venait d'être crée, nous le l'utilisions pas encore. Il aura plus de succès que les noms binon (bit) et bogue (bug) créés en même temps.

31. « Virtuelle » signifie qu'il ne s'agit pas d'un vrai ordinateur mais d'un ordinateur fictif émulé par un logiciel.

machine C. Le tout est écrit dans un autre langage d'assembleur. C'est un processus un peu fastidieux, car le langage n'est pas convivial, mais réaliste, quoique, dans un but didactique, il soit nettement simplifié par ses concepteurs, par rapport à un langage d'assembleur réel de l'époque. Il n'en demeure pas moins que cette activité m'a beaucoup appris. Avant l'arrivée du CII 10070, nous avons la possibilité de passages sur la CAE 510, la seule machine qui nous est accessible. Pour cela, il faut écrire le programme sur des bordereaux ad hoc, puis transmettre ces bordereaux à une perforeuse [32] qui nous rend dans un casier un paquet de cartes ; puis munis de notre paquet de cartes, nous pouvons accéder à la CAE 510, dans une salle spéciale qui lui est dédiée. Auparavant, il faut réserver la salle. Mon groupe de projet est composé de Roger Mohr [33] et de Marcel Navet [34]. Le seul créneau disponible, que nous voulons suffisamment long pour pouvoir faire plusieurs passages en cas d'erreurs, est la nuit du samedi au dimanche. L'un d'entre nous a retiré, auprès de la secrétaire, la clé du centre de calcul qui nous permettra de rentrer et sortir nuitamment. Nous commençons le long processus : d'abord en démarrant la CAE510 en positionnant les switchs du mot sur lequel l'ordinateur boote. Le programme qui démarre l'ordinateur est sur une bande magnétique, donc nous n'avons rien à faire qu'attendre que la phase de démarrage soit complète. D'ailleurs ≪ attendre ≫ est l'une de nos principales activités. Pour le compilateur du langage dans lequel la machine C est simulée, il nous faut le charger, car il est sur un ruban perforé. Le lancement de la lecture se fait sur la machine à écrire IBM à boule qui est la console de contrôle opérateur. Les différentes phases du processus sont décrites dans un classeur qui doit rester toujours près de l'ordinateur, sur une table. La plupart du texte est dactylographié, mais des modifications ont été ajoutées au stylo à bille après qu'ont été raturées les consignes obsolètes. Nouvelle attente ! Puis sur une nouvelle instruction à la console, la CAE engloutit notre paquet de cartes perforées. Nouvelle attente ! La machine simulée commence à lire notre programme (nos données), puis se met à calculer. Chaque phase de calcul a son propre bruit. Avant la lecture de la bande magnétique c'est relativement silencieux, seul le clignotement des diodes rouges montrent que l'ordinateur calcule. Quand vient son tour, on entend surtout le léger son du dispositif qui la maintient tendue, en gros un petit tchak toutes les demies secondes. La lecture du ruban perforé s'apparente à un sifflement. Puis vient le tacatac régulier de la lecture des cartes perforées [35]. Zut, il y a une erreur ! Nous discutons pour en trouver l'origine et proposer une suggestion de correction. Quand nous la localisons, nous utilisons la perforatrice mise à la disposition du public pour perforer une ou plusieurs nouvelles cartes. C'est reparti jusqu'à la prochaine erreur ! Vers onze heures du soir, par inadvertance, je tape sur le clavier de la console (l'IBM à boule). L'ordinateur se plante et le processus que nous avions entamé vers 19h, doit être recommencé (boot, chargement du compilateur, lecture du programme de simulation de la machine virtuelle, lecture de nos cartes, premiers calculs). Mes condisciples, après avoir râlé, préfèrent en sourire. Cela aurait pu leur arriver, à ce qu'ils prétendent. Au lieu de quitter le centre de calcul vers minuit, dans une prévision optimiste, nous abandonnons les lieux vers 2h du matin. Notre assembleur prototype traduit presque toutes les instructions. Nous sommes contents. Quelques petites mises au point,

32. Une ≪ perforeuse ≫ est une employée du centre de calcul. Sans paraître sexiste, il faut avouer que cette fonction est toujours occupée par une femme. Elle perfore (fait des trous dans) les cartes en se servant d'une énorme machine bruyante, qui s'appelle une ≪ perforatrice ≫.

33. Roger Mohr a été professeur à l'École des Mines de Nancy, puis à l'Ensimag de Grenoble dont il deviendra le directeur [50].

34. Il deviendra chargé de recherche au CNRS en physique à Orléans et son fils Nicolas Navet est professeur d'informatique à l'Université du Luxembourg.

35. Les imprimantes à tambour viendront plus tard et, trop bruyantes, seront installées dans des pièces séparées. Elles ont le *po-pom po-pom* qui les caractérise, qui devient très rythmé quand elles impriment un dump mémoire pour un chercheur qui n'a pas pu identifier la source de son erreur et utilise l'ultime recours pour traquer ce que nous n'appelons pas encore un bug.

avec un meilleur créneau horaire, permettront de rendre un rapport et un projet acceptables. En ce qui nous concerne, comme nous ne sommes pas de véritables étudiants, c'est plutôt pour la gloire !

C'est aussi à cette époque que je suis des cours de Claude Pair sur les automates finis, les langages à contexte libre, les grammaires et l'analyse syntaxique. A ce cours est associé un polycopié qu'a rédigé Alain Quéré [36] et qui correspond à des notes qu'il a prises lors d'un cours précédent [43]. C'est à cette occasion que j'apprends l'importance du point-fixe en informatique [37]. Je suis aussi des cours sur les graphes et les algorithmes de cheminement, théorie dans laquelle Claude Pair et Jean-Claude Derniame ont joué un rôle de pionniers et où aussi le point-fixe est fondamental [13, 12, 1].

FIGURE 4 – Une imprimante à tambour, capot ouvert. *Source Wikimedia Commons.*

3 Ma thèse de troisième cycle

Pour Claude Pair, les problèmes centraux de l'informatique naissante sont l'analyse syntaxique et les automates. Il s'intéresse aussi aux arbres (syntaxiques) dont la structure ressemble par certains aspects aux chaînes de caractères, en les généralisant, et qui jouent un rôle fondamental en analyse syntaxique. On doit pouvoir donc décrire des automates d'arbres et définir des langages d'arbres reconnus par des automates d'arbres, que l'on appelle langages reconnaissables. D'autre part, un théorème dit que, pour les chaînes de caractères, les langages *reconnaissables* par automates sont aussi ceux qui sont engendrés par les opérations algébriques de bases sur les langages dont la fameuse opération *étoile*, pour cette raison on les appelle langages *réguliers* ; ils sont aussi ceux qui sont solutions d'équations d'un certain type et pour cela ils sont *algébriques*. Claude [38] veut donc que je généralise ce résultat aux arbres. Mais il se trouve que je suis tombé sur des articles qui proposent une généralisation encore plus grande, en permettant d'énoncer ce théorème sur des structures plus générales, à savoir des algèbres abstraites, dont les chaînes de caractères et les arbres ne sont que des cas particuliers et plus

36. Alain Quéré à été maître de conférence à Nancy. Il deviendra directeur de l'Unité Inria-Lorraine, puis directeur du Centre informatique national de l'enseignement supérieur (CINES) à Montpellier.

37. Le point-fixe est au cœur du livre de Livercy [39], car il est le pivot de la récursivité.

38. À l'époque, je ne me serais pas permis de l'appeler par son prénom.

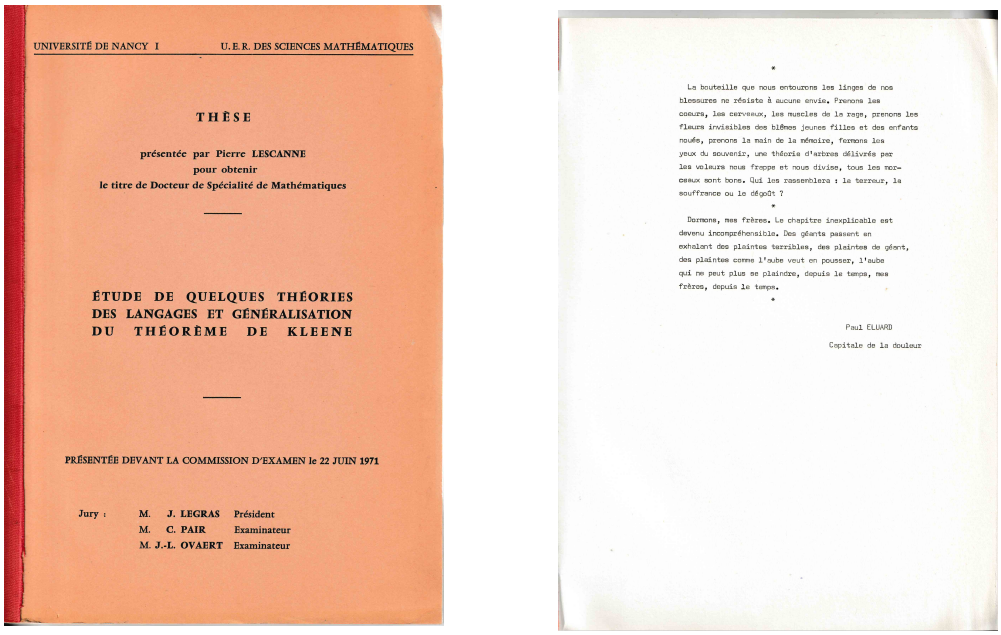
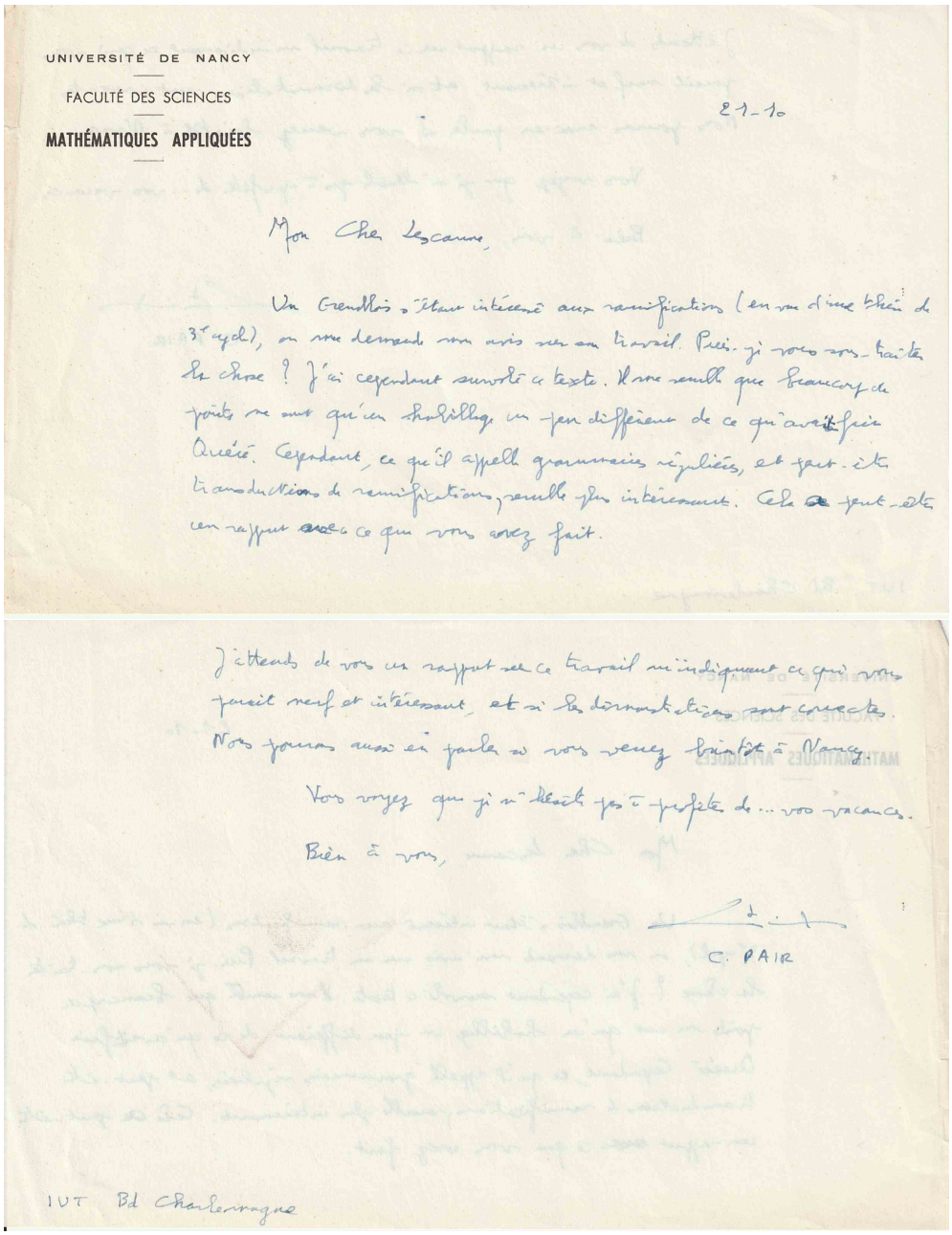


FIGURE 5 – La couverture de ma thèse de troisième cycle et la citation d'Éluard

généralement encore en prenant comme cadre la théorie mathématique des catégories, si bien que, par ironie, je mets en exergue de ma thèse [31] un poème d'Éluard [39], avec en particulier les trois extraits « *une théorie d'arbres [...] nous frappe et nous divise* », « *Le chapitre inexplicable est devenu incompréhensible* » et « *Prenons la main de la mémoire* » [40] qui s'appliquent bien, je pense, à mon travail. De fait, je peux généraliser l'étoile et démontrer un théorème d'équivalence des reconnaissables, des réguliers et des algébriques [45]. Ce résultat n'a pas d'écho. Plus tard, en 1976, je publie (en français), dans la revue d'informatique française *RAIRO Informatique théorique*, un article [33] qui n'est cité, d'après google scholar, que trois fois. Ainsi va la recherche en informatique française dans les années 1970 ! En arrivant au MIT en 1980, je constate que la recherche française en informatique est insignifiante pour nos collègues américains, tandis qu'en France, en informatique théorique, il existe un rivalité Paris-Province.

En 1971, c'est le moment pour moi de faire mon service militaire. Étant agrégé, je peux bénéficier de mon statut pour l'effectuer comme professeur de mathématiques à l'École de Maistrance de l'Aéronautique navale à la base de Fréjus-Saint-Raphaël. Je suis logé sur la base et pour aller de ma chambre à la plage, il me faut contourner le tennis, ça aurait pu être pire ! J'assure quelques cours, mais quand arrive un contingent d'agrégés et de certifiés, matelots sans spé, je distribue ces cours à mes collègues, qui ainsi peuvent se protéger d'un second maître intrusif en prétendant, à juste titre, qu'ils préparent leur enseignement. Certains de nos élèves, futurs navigants de l'Aéronavale sont recrutés au niveau du bac, parmi des étudiants en réorientation après une première expérience à l'Université et reçoivent ici une formation censée leur donner un petit vernis scientifique et surtout leur conférer l'esprit de la grande

39. Paul Éluard, *Silence de l'Évangile* dans le recueil, *Mourir de ne pas mourir*. Le titre de cet article, « *Prenons la main de la mémoire* », a été extrait de ce poème.
40. Ici au sens informatique du terme.



UNIVERSITÉ DE NANCY

FACULTÉ DES SCIENCES

MATHÉMATIQUES APPLIQUÉES

27-10

Mon Cher Lescanne,

Un Grenoblois s'étant intéressé aux ramifications (en vue d'une thèse de 3e cycle), on me demande mon avis sur son travail. Puis-je vous soumettre la chose ? J'ai cependant survolé le texte. Il me semble que beaucoup de points ne sont qu'un charabia un peu différent de ce qu'avait fait Queiné. Cependant, ce qu'il appelle grammaires régulières, est peut-être traduction de ramifications, semble plus intéressant. Cela se peut-être en rapport avec ce que vous avez fait.

J'attends de vous un rapport sur ce travail m'indiquant ce qui vous paraît neuf et intéressant, et si les démonstrations sont correctes. Nous pourrons aussi en parler si vous venez bientôt à Nancy.

Vous voyez que j'ai bien tort de profiter de vos vacances.

Bien à vous,

C. PAIR

IUT Bd Charlemagne

FIGURE 6 – Une lettre de Claude Pair, reçue pendant mon service militaire

arme dans laquelle ils vont servir. Ensuite, ils suivront une formation avec les « aviateurs », qui toiseront ces « matelots » qui ne sont ni complètement marins, ni complètement aviateurs. Je me suis mis en tête de leur enseigner les principes de la *plate-forme à inertie* (le nec plus ultra de l'instrument de navigation avant le GPS). Pour cela, je leur explique les principes du gyroscope,

grâce à ce qui me reste de mon cours de *méca géné*[41]. Je termine ma théorie (très adoucie) de la précession gyroscopique par la manipulation d'un vrai gros gyroscope de démonstration extrait des armoires de l'École. Le cours passe moyennement. Mais il passe quand même, car quelques mois plus tard, je revois mes élèves en formation chez les « aviateurs » ; ils me remercient de mon cours, car quand celui sur les plates-formes à inertie a lieu, les « aviateurs » peinent sur le concept paradoxal de précession, mais mes anciens élèves, ceux de l'Aéronavale, peuvent frimer : ils tiennent leur revanche. En fait, cette année là ressemble à des vacances, sauf que je suis loin de ma famille, en effet, mon fils aîné a vu le jour en mars 1971. Mais comme il y a un train de nuit direct de Saint-Raphaël à Nancy et que les billets de train sont soit gratuits, soit à 75% de réduction (je ne suis pas payé), je peux aller les voir souvent, quand ma femme et mon fils ne viennent pas à Fréjus pour les vacances. Mon épouse doit cependant s'occuper seule de notre jeune fils et faire bouillir la marmite. Dans cette ambiance fréjussienne qui mélange le loisir et la Marine, ce qui me manque aussi beaucoup, c'est l'environnement scientifique. Je comble cela en continuant une recherche à la suite de ma thèse, mais celle-ci s'élève dans l'abstraction. Claude Pair en particulier m'envoie des documents à lire, en l'occurrence une thèse à évaluer (figure 6).

Heureusement, il est prévu une école d'été d'informatique (la deuxième du genre) à Neuchâtel en Suisse. Comme je l'ai dit dans l'introduction, ces écoles d'été qui ont été créées par Claude Pair sont cruciales dans notre formation. J'ai raté la première édition, qui a eu lieu à Alès et je le regrette[42]. Au cours de cette école, au demeurant très sympathique et familiale, les jeunes chercheurs sont invités à présenter leurs travaux récents. J'expose les miens, mais je ne sais pas bien m'expliquer, je ne donne aucune intuition, je m'envole dans l'abstraction. Éluard a raison et, au premier rang de l'assistance, l'un de mes meilleurs amis pleure (littéralement) de rire, tant ce que je dis est abscons. Il me faut atterrir. Bien que la théorie des catégories reste bien en cour en informatique théorique, je n'y toucherai plus, à deux petites exceptions près qui se révéleront décevantes. En plus d'être incompréhensibles, les catégoriciens de l'informatique théorique partagent avec les économistes leur sectarisme[43]. Cependant, je ne sais pas pourquoi, peut-être pour mon regard averti, mais circonspect sur cette théorie, on m'a demandé lors du 11$^{\text{th}}$ Workshop on Specification of Abstract Data Types, en 1996, d'animer une table ronde sur *l'intérêt de la théorie des catégories en informatique*. De ce que j'en retiens, cette table ronde ne nous a pas beaucoup appris sur le fond du problème, mais beaucoup sur la sociologie des mathématiciens[44].

Au même moment, c'est-à-dire en 1972, je rencontre une théorie qui va jouer un rôle important dans ma recherche ultérieure, ainsi que dans mon enseignement[45]. En effet, je découvre le livre de Peter Wegner intitulé *Programming languages, information structures, and machine organization* [53] qui, entre autres choses, me fait connaître le *lambda-calcul*. Pour compléter mon initiation, je fais acheter par la bibliothèque de mathématiques le petit fascicule dactylographié de Hindley, Lercher et Seldin [19], qui, bien que ce ne soit pas dans son titre, présente plus complètement le lambda-calcul. C'est aussi à ce moment que Claude Pair rapporte d'une

41. Le certificat de Mécanique Générale, très mathématique, lui aussi, malgré son nom.

42. J'assisterai à celle de Tarbes en 1974 et à celle de Rabat en 1975.

43. Que la rationalité des agents ne les conduise pas forcément à une situation d'équilibre heurte les spécialistes de théorie des jeux, parce qu'il fragilise leur dogme et leur foi en la croissance. En conséquence, ils rejettent un résultat pourtant formellement démontré [38].

44. J'y apprends que la théorie des catégories est prétendument une théorie d'extrême gauche. Le titre du livre de référence de Mac Lane, destiné au « mathématicien travailleur » serait une allusion ironique à cet aspect [25].

45. A l'École normale supérieure de Lyon, à partir de 1997, j'enseignerai le lambda-calcul à des générations d'élèves.

réunion du groupe de travail de l'IFIP [46], les photocopies des transparents de Dana Scott [47] sur le modèle du lambda-calcul qu'il vient d'inventer [48]. Je suis fasciné et pour faire partager mon enthousiasme, j'écris, en français, pour mes collègues, une petite note sur le sujet [32]. Je ne renouerai avec le lambda-calcul qu'en 1993 [36].

Pendant mon service militaire, j'ai été promu maître assistant, puis nommé à l'Université Nancy 2, l'université de lettres et droit. Je continue à enseigner les mathématiques, cette fois-ci aux psychologues. Quant au contenu, c'est assez cool et avec mon collègue Jean-Pierre Deschaseaux, enseignant en mathématiques, nous essayons d'intéresser les étudiants, qui sont réfractaires aux mathématiques. Nous nous investissons donc beaucoup dans la pédagogie, mais lorsqu'un collègue enseignant de psychologie se lamente du grand nombre d'étudiants eu égard aux perspectives professionnelles et nous invite à être en pointe pour la sélection, car les maths sont un bon outil pour cela, alors que la psychologie en tant que discipline d'enseignement ne peut pas jouer ce rôle, nous opposons un refus catégorique. Sans le lui dire, mais en le faisant savoir aux étudiants, nous mettons au point un calcul savant de coefficients, pour qu'un étudiant qui a assisté à tous ses partiels, sans cependant remettre jamais une copie complètement blanche, soit assuré d'avoir la moyenne. Un jour que les étudiant(e)s sont en grève, je vais prendre un café avec le piquet de grève et nous discutons de tout et de rien, notamment de l'intérêt des mathématiques dans le cursus de psychologie. Une étudiante qui, je le comprends bien, n'est pas enthousiasmée par mes cours, me le dit de façon cachée : « Oh Monsieur vous devriez assister au cours d'histoire du cinéma, c'est formidable ! » [48] Je n'enseigne toujours pas d'informatique, ni quelque chose qui y ressemble, mais un dossier pour la création d'une MIAGE [49] à Nancy est déposé. On me demande d'écrire le programme de mathématiques et je m'y colle, puis je me retrouve à l'enseigner devant la première promotion : graphes et langages entre autres, mon enseignement se rapproche de l'informatique, puisque j'aborde enfin les structures mathématiques de l'informatique. Dans la plaquette de présentation j'essaie de trouver les mots qui puissent motiver des étudiants de sciences économiques à rejoindre notre nouvelle formation [15].

L'enseignement des mathématiques doit se fixer trois objectifs :

- *Donner des outils qui serviront soit à d'autres enseignements, soit dans la formation permanente, soit dans la vie professionnelle.*
- *Introduire la démarche algorithmique et la modélisation de situations concrètes.*
- *Développer l'esprit de rigueur dans la formulation et la résolution des problèmes.*

On voit que plus que le contenu, c'est la démarche qui compte. C'est aussi une initiation à l' « algorithmique ».

En parallèle, je donne des cours au département d'informatique de l'IUT de Nancy, toujours des cours de mathématiques, toujours dans le même esprit, mais qui incluent l'algèbre de Boole et les fameux diagrammes de Karnaugh, que j'apprends pour l'occasion.

4 Attaché de recherche

En 1973, mes multiples charges d'enseignement sont lourdes et, comme nous venons de le voir, changent sans arrêt. D'autre part, l'équipe de recherche de Claude Pair est reconnue par le CNRS, nous sommes une ERA, ce qui signifie *équipe de recherche associée*. Claude Pair invite

46. L'IFIP (*International Federation for Information Processing*) regroupe au niveau international, les sociétés savantes en informatique.

47. Dana Scott est grand logiciel américain qui s'est intéressé aux fondements de l'informatique.

48. Le cours est enseigné par Roger Viry-Babel, un an plus âgé que moi, avec qui je ne prétends pas rivaliser sur la forme comme sur le fond. Grand cinéphile, Roger Viry-Babel (1946-2006) fut aussi cinéaste, journaliste et pionnier de l'enseignement de l'audiovisuel à l'université.

49. Maîtrise d'informatique appliquée à la gestion des entreprises.

FIGURE 7 – Une machine à écrire IBM à boule. *Etan J. Tal via Wikimédia Commons.*

ses chercheurs à candidater sur un poste d'attaché de recherche, ce que je fais. Donc en 1974, je suis intégré au CNRS et je peux, en étant déchargé de mes tâches pédagogiques, me consacrer entièrement à la recherche. Je deviens donc « détaché attaché agrégé ». Le CNRS m'attribue un « parrain », c'est-à-dire un chercheur confirmé qui à côté de mon directeur de recherche me guide dans mes premiers pas de chercheur. Ce parrain est Jean-François Perrot [50] ; il me donnera de précieux conseils ; en particulier il m'apprendra à faire une bibliographie scientifique en respectant la « norme », concept dont j'ignorais l'existence et dont je me souviendrai toute ma carrière. Il deviendra un ami et nous ne nous retrouverons que dans des lieux exotiques, à Belmont Massachusets, où il logera chez nous, et à Nagoya et Kyoto au Japon, où nous logerons les uns chez les autres. Pour commencer dans ma nouvelle fonction, j'écris l'article issu de ma thèse de troisième cycle [33], signalé dans la première partie. Suite à l'*école avancé d'informatique théorique* d'Amsterdam [51] (section 5) et influencé par le μ-calcul de Jaco de Bakker [11], je m'intéresse au calcul relationnel, c'est-à-dire à un calcul dans des structures mathématiques où la notion de base est la relation binaire [52]. Ces calculs sont vite inextricables, je décide de les automatiser et je choisis le langage de haut niveau qui me parait le plus adapté, à savoir Pascal. En effet, ce langage a une belle syntaxe, dérivée d'Algol 60, offre des structures de données et il y a sur l'IRIS 80, le nouvel ordinateur du centre de calcul [53], un compilateur opérationnel. Cet ordinateur est localisé au Centre de calcul qui a déménagé au château du Montet (figure 8), riche bâtisse du XIX$^{\text{ème}}$ siècle entourée de son immense parc [54], qui fut autrefois le résidence des « maîtres de forges » Fould, propriétaires des Aciéries de Pompey. Ce château est situé sur une éminence qui domine Nancy et donc pour y aller en bicyclette, il faut avoir une bonne forme physique. La première année, mon programme est sur cartes perforées, mais un jour, je franchis le pas et je passe à la console. Mon programme réside alors dans l'une des mémoires de l'ordinateur et j'interagis avec lui à l'aide d'un télétype ASR 33 (figure 11), qui est un pupitre devant lequel je suis assis et sur lequel il y a un clavier de machine à écrire et un rouleau de papier où une imprimante permet de voir ce que l'on tape et ce que l'ordinateur nous répond. Tout cela est assez bruyant. Je modifie les programmes avec l'éditeur de texte ligne à ligne qui ressemble à "ed" d'UNIX ou "edit" de MS-DOS. La

50. Jean-François qui était à l'époque (1974) versé dans la théorie des langages est devenu un spécialiste des systèmes multi-agents et des langages à objets.

51. A partir de 1974, Jaco de Bakker avait organisé biennalement des écoles destination des jeunes chercheurs en informatique fondamentale, intitulées *Advanced Course on Foundations of Computer Science.*

52. Chapitre 5 de ma thèse d'État [34]

53. L'IRIS 80 a 1 Mo de mémoire centrale et 500 Mo de mémoire périphérique [49, 17].

54. Dont une partie est maintenant le jardin botanique Jean-Marie Pelt.

différence – elle est de taille – est que le programme est bogué (dû, très probablement, à une erreur de programmation sur laquelle nous mettons en garde nos étudiants débutants qui apprennent à programmer des listes). Quand on se promène vers la fin d'un fichier, l'éditeur peut se planter sur une instruction d'avancement ou de retour arrière. Je n'ai jamais identifié la séquence d'instructions qui aboutit au crash, car je suis concentré sur mon programme et je n'ai pas cherché un scénario de crash. D'ailleurs, je ne sais pas si l'identification de la séquence coupable m'aurait permis ultérieurement de l'éviter. Ce que je sais, c'est qu'en cas de plantage le fichier que je manipule est irrémédiablement perdu, car, autre caractéristique de l'éditeur ligne à ligne de l'IRIS 80, il travaille directement sur le fichier, pas sur un tampon. Si je n'ai pas fait auparavant de sauvegarde incrémentale (c'est-à-dire un enregistrement sur un fichier qui est à chaque fois nouveau) et que l'éditeur se plante, il ne me reste plus que mes yeux pour pleurer. Mais parfois, ce n'est pas seulement l'éditeur qui se plante, mais l'IRIS 80 entier. À l'exception de quelques jurons, le silence se fait alors dans la salle, puisque toutes les ASR 33 s'arrêtent en même temps. Nous allons d'abord aux nouvelles, puis s'il fait beau et que la panne semble être de courte durée, nous sortons dans le parc pour discuter et faire connaissance, car après tout, nous venons de différents horizons de la science et nous n'avons pas l'occasion de nous parler, à part bonjour–au-revoir. Un jour, les ingénieurs nous ont suggéré de rentrer chez nous, car la panne n'était pas anodine : elle a duré trois mois ! Donc, pour prévenir ces aléas, je me retrouve avec des fichiers dont les noms se terminent par 253, puis 254, puis 255, etc. et l'enregistrement systématique et incrémental de mes fichiers devient un réflexe que j'ai toujours gardé, malgré les systèmes sophistiqués de sauvegarde que nous avons.

Dans ce cadre, je programme un petit système de manipulation symbolique, comprenant, entre autres, un algorithme d'unification du premier ordre [55] dont je suis assez content. Les mathématiques expérimentales ont toujours eu ma faveur, à savoir découvrir une propriété mathématique à partir d'expériences sur ordinateur. Ici il s'agit de voir comment un ordinateur se débrouille avec des égalités, en fait il se débrouille mal, si les égalités ne sont pas orientées. Ces résultats sur les démonstrations en logique relationnelle des égalités forment le chapitre 6 de ma thèse d'État soutenue en 1979 [34] et m'aideront beaucoup quand je programmerai mon logiciel REVE [35], lors de mon séjour au MIT, mais aussi ils me font prendre conscience que la manipulation d'égalités ne peut pas être rendue efficace [56] et ce constat me conduira à la réécriture.

Pendant cette période je ne fais pas ou très peu d'enseignement. Cependant Claude Pair qui assure un cours de théorie des langages et de principes de la compilation à l'École des Mines de Nancy me demande d'assurer les travaux dirigés. C'est une expérience d'enseignement très frustrante. La direction de l'École a du mal à motiver les élèves et n'a trouvé que la feuille d'émargement comme seul moyen d'assurer l'assiduité. Ceux-ci ne viennent en travaux dirigés que pour la signer. Ils bavardent tout le temps, ne m'écoutent pas, ne font pas les exercices que je leur demande. Ils n'ont aucun goût pour les cours de « babasse ». Je n'ai pas l'autorité pour les faire participer. C'est d'autant plus triste que certains deviendront des décideurs ou des chefs de services informatiques sans avoir jamais rien appris de cette discipline. À leur décharge, il faut avouer que le discours ambiant de l'époque affirme que l'informatique n'est réservée qu'à des cercles restreints. Elle attire d'autant moins les élèves que les autres disciplines la dénigrent. L'équipement en ordinateurs n'est pas à la hauteur, si l'on se réfère aux États-

55. Mon algorithme d'unification est semblable à celui de Gérard Huet [20], que j'ai retrouvé plus ou moins indépendemment. J'ai découvert l'unification en 1974 à l'école de printemps d'Amsterdam par Bob Kowalski [24], puis, plus tard, j'ai suivi un séminaire de Gérard Huet. En fin de compte j'ai reconstruit mon propre algorithme pour mon programme.

56. Au moment (début de l'année 2021) où j'écris ces lignes des chercheurs de mon équipe de recherche lyonnaise rapportent le même constat qui est maintenant bien établi [40].

FIGURE 8 – Le château du Montet,
photographie de Nicolas Fressengeas disponible sur Wikimédia Commons

FIGURE 9 – Croquis de l'IRIS 80

Unis comme je pourrai le constater. Les élèves ont épisodiquement accès par cartes perforées à un ordinateur situé au dernier étage du bâtiment ; pas d'écrans, pas de terminaux, pas de connexion interactive ! A Nancy, qui a pourtant beaucoup d'écoles d'ingénieurs, l'informatique n'a pas commencé dans l'une de ces écoles, mais à la faculté des sciences. Ce phénomène s'est

reproduit dans presque tous les sites universitaires français et n'est pas fortuit. Les grandes écoles ont toujours beaucoup de mal à accepter une nouvelle technologie. L'informatique y est aujourd'hui en odeur de sainteté. Est-ce un bon présage ?

À cette époque déjà l'informatique effraie certains, mais je ne pense pas que cette peur affecte mes élèves de l'École des Mines, peu concernés. L'emprise d'une multinationale qui dominerait tous les moyens de communication fait courir de grands risques à la liberté et à la démocratie. Deux sociétés sont dans le viseur : IBM dont un livre que je lis dénonce les turpitudes [41] et ITT qui soutient Pinochet dans son coup d'état au Chili. En parallèle, je lis deux livres de Noam Chomsky, son livre *Structures syntaxiques* [3] et son livre *L'Amérique et ses nouveaux mandarins* [2]. Quand plus tard je serai au MIT, j'assisterai à une de ses conférences, très politique, dans l'esprit du deuxième sujet.

Il n'y a pas que les équipements de l'École des Mines qui sont pauvres. Ceux mis à la dispositions de tous les étudiants et de tous les chercheurs le sont aussi. Nous ne nous rendons pas compte de cette situation, faute de comparaisons, mais je le constaterai crûment lors de mon séjour au MIT, en 1980. En France, on nous vante le Mitra 15, mais il n'a pas de logiciel. Quant à l'IRIS 80, c'est un dinosaure sans puissance de calcul, nécessitant un personnel pléthorique et possédant un pauvre système d'exploitation, comme cela est bien reflété par la figure 3. Cette déplorable situation est due bien sûr au Plan Calcul qui nous isole du reste du monde. Nos chercheuses et chercheurs peuvent inventer une nouvelle architecture de machine ; il suffit d'une petite équipe pour cela. Mais munir cet ordinateur de logiciels décents suppose une masse critique qui ne peut avoir lieu qu'à l'échelle d'un pays comme les États-Unis ou d'un effort international. L'effort de créer un logiciel vaut le coup (le coût) que si l'on sait qu'il atteindra un nombre important d'utilisateurs. Le différentiel d'équipement matériel et logiciel est illustré par la figure 11. Cette situation a aussi pour autre raison, l'incurie de nos décideurs. Le polytechnicien qui dirige la France à l'époque n'est conseillé que par des amis, avec la même culture que celle que je rencontre à l'École des Mines. Ces amis ne croient qu'aux PTT (portant bien obsolètes en France [57], malgré ou à cause de son corps d'ingénieurs de l'État) et à la grosse production industrielle qui a fait le succès des trente glorieuses et de la Lorraine. J'assiste, au château du Montet, à une démonstration de Cyclades [30] (le réseau d'ordinateurs précurseur d'Internet) et j'écoute une conférence de Louis Pouzin, son créateur. Je suis consterné quand j'apprends l'arrêt de ce projet. Encore une occasion ratée ! Fini l'espoir d'une coopération interrégionale ou internationale !

5 Tarbes et Livercy

En mai 1974, avec Jean-Pierre Finance [58] et Jean-Luc Rémy, nous assistons, à Amsterdam, à la première des écoles de printemps en informatique fondamentale, organisée par le Mathematisch Centrum [59]. Pour l'anecdote, je me rappelle très bien que deux étudiants hollandais accaparent presque toutes les questions lors de sessions après les cours. Ils m'impressionnent et m'énervent parce qu'ils semblent avoir tout compris et montrent qu'ils maîtrisent parfaitement bien l'anglais. Ils sont devenus par la suite de brillants chercheurs. Je comprends qu'il faudra que je progresse en anglais si je veux faire de la recherche au niveau international. Cette école est une véritable formation à l'informatique fondamentale. Il se trouve que l'École d'été

57. Je n'ai eu le téléphone à la maison qu'en 1975 à condition de payer plus d'un an d'avance de communications téléphoniques. Un de mes frères vivant en plein Morvan n'a eu l'automatique que 1982.

58. Jean-Pierre Finance a été président de l'Université Henri Poincaré, puis président fondateur de l'Université de Lorraine.

59. Jean-Luc Rémy et moi apprécions tellement l'école de 1974 que nous participons aussi à celle de 1976.

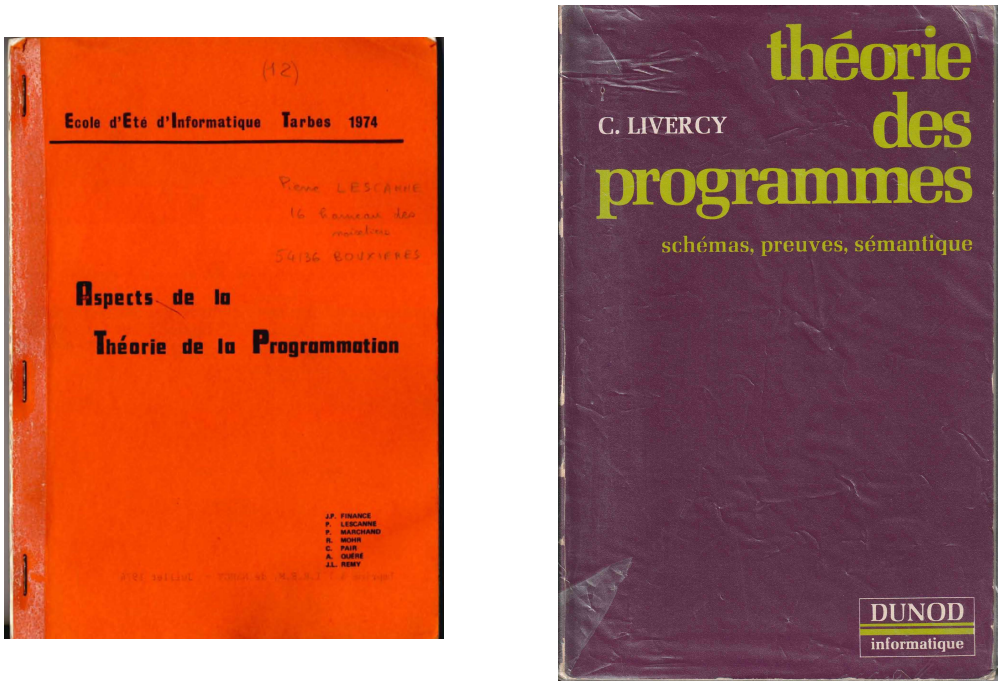


FIGURE 10 – Polycopié de l'École d'été d'informatique de Tarbes en 1974 et livre de C. Livercy. *Photographies de l'auteur*

de l'AFCET [60], plus généraliste par nature, propose cette année 1974, à Claude Pair et à son équipe, d'enseigner un cours d'informatique fondamentale que nous décidons d'appeler *Aspects de la théorie de la programmation* [16] pour lequel nous réalisons un polycopié (figure 10). Nous y présentons l'état de l'art en ce qui concerne l'analyse et la sémantique des programmes en nous inspirant bien sûr de ce que nous avons appris à Amsterdam. Ce cours donne naissance par la suite à un livre publié chez Dunod [39] signé de Livercy, nom de plume commun des auteurs [61] et intitulé *Théorie des programmes* (figure 10). Nous changeons *programmation* en *programmes* quand nous nous rendons compte que notre étude porte finalement sur les programmes et non pas sur la programmation.

6 Le CRIN

En 1976, l'ERA devient *Laboratoire Associé* (LA) et prend le nom de *Centre de recherche en Informatique de Nancy* (CRIN). Il se structure en équipes. Je rejoins l'équipe de recherche sur la programmation qui s'appelle CASTOR. En 1979, nous accueillons Jean-Pierre Jouannaud [62] qui vient d'être nommé professeur et j'encadre mon premier étudiant en thèse, Fernand Reinig [63] dont Jean-Pierre reprendra la direction de thèse après mon départ aux États-Unis. Suite à une

60. L'AFCET est la société savante de l'informatique de l'époque.
61. Jean-Pierre Finance, Monique Grandbastien, Pierre Lescanne, Pierre Marchand, Roger Mohr, Alain Quéré et Jean-Luc Rémy.
62. Jean-Pierre Jouannaud a été ensuite professeur à l'Université d'Orsay.
63. Fernand Reinig a présidé le LIST (*Luxembourg Institute of Science and Technology*).

école de printemps à Sophia Antipolis en 1979, où j'ai assisté à des exposés de Gérard Huet [64] et Dallas Lankford, je me suis passionné pour la réécriture [65] et je transmets mon enthousiasme à Jean-Pierre. Nous formerons l'équipe Euréca [66]. Michel Sintzoff [67] est présent au laboratoire sur un *poste rouge* du CNRS [68]. Je parle à l'un et à l'autre de la procédure de Knuth et Bendix [23] qui est au cœur de la théorie. Pour en savoir plus, Michel écrit à Donald Knuth [69] et obtient un tiré à part, sur lequel nous travaillons. Pour ma part, je m'intéresse à la démonstration de la terminaison des systèmes de réécriture (autrement dit, la démonstration que toutes les simplifications possibles se terminent). Or Claude Pair va régulièrement à l'IRIA à Rocquencourt et il en rapporte régulièrement une énorme brochure qui est la photocopie des listes de tous les rapports internes récents des laboratoires de recherche en informatique étasuniens. Je les épluche pour savoir ce qui peut bien se faire en recherche en informatique. Un jour, dans la liste de celui de l'Université d'Illinois à Urbana-Champaign, dont j'ignore totalement l'existence, je découvre un rapport qui peut m'intéresser [46]. Je prends ma belle plume et dans mon meilleur anglais possible j'envoie, par la poste, une carte au service de documentation du laboratoire d'Urbana-Champaign et, trois mois plus tard, je reçois une enveloppe contenant le rapport demandé. David Plaisted [70], son auteur, est très brillant et inventif, mais n'est pas le meilleur pédagogue qui soit. Au prix d'un effort de plusieurs semaines, sinon de mois, je déchiffre l'idée qui me parait fantastique et initie ma recherche sur la terminaison des systèmes de réécriture ; plus tard, je sympathiserai avec ce chercheur. En le simplifiant et en le rendant compréhensible pour mon étudiant Fernand Reinig, je découvrirai une version d'un autre ordre de terminaison qui sera à l'origine non seulement d'une collaboration, mais aussi d'une amitié avec son auteur Nachum Dershowitz [71] [14].

7 Le MIT

En 1980, j'obtiens du CNRS l'autorisation de pouvoir passer une année au MIT au *Laboratory for Computer Science* à l'invitation du Professeur John Guttag, dont l'équipe est très liée à celle de Barbara Liskov (prix Turing 2008). J'y reste une deuxième année, donc, en tout, j'y suis d'août 1980 à août 1982. J'y rencontre l'informatique, la vraie, celle des logiciels et des matériels conviviaux et complexes qui fonctionnent véritablement et connectent les gens, dont je n'ai jusque là entrevu que quelques minuscules aspects. En fait, en arrivant au LCS (Laboratory for Computer Science), je passe directement de l'Âge de la pierre aux Temps modernes ou plus précisément des cartes perforés et du traitement par lot, aux postes individuels, et aux ordinateurs avec partage d'accès, efficaces et simples d'utilisation ainsi qu'aux Alto de chez Xerox qui sont les premières stations de travail modernes avec écrans bitmap et souris [72]. J'y apprends à me servir d'Emacs et d'UNIX que j'utilise encore aujourd'hui. Je programme en CLU dont OCAML ou HASKELL ont repris les concepts. Surtout, j'ai accès à un terminal avec

64. Gérard Huet est membre de l'Académie des Sciences
65. La réécriture est un domaine de recherche qui se situe entre le calcul symbolique et le déduction automatique. Elle étudie comment des expressions mathématiques peuvent être ≪ simplifiées ≫ de façon efficace.
66. Nous obtiendrons la médaille d'argent du CNRS en 1987.
67. Michel Sintzoff est un chercheur belge en informatique, compagnon de route de Claude Pair.
68. C'est ainsi que l'on appelle les postes de professeur invité du CNRS.
69. Donald Knuth reste, encore aujourd'hui, **la** référence en informatique en ce qui concerne l'algorithmique.
70. David Plaisted a passé sa thèse à l'Université de Stanford, puis à été professeur associé à l'Université d'Illinois à Urbana-Champaign et professeur titulaire à l'Université de Caroline du Nord à Chapel Hill.
71. Nachum Dershowitz a passé sa thèse au Weizmann Institute en Israël, puis a été professeur associé à l'Université d'Illinois à Urbana-Champaign, puis professeur titulaire à l'Université de Tel-Aviv.
72. Encore fragiles et peu nombreuses, je ne les ut liserai pas de façon routinière.

FIGURE 11 – Un télétype ASR 33. *Source Wikimedia Commons* et un terminal Zenith. *Jamie Cox from Melbourne, USA, CC BY 2.0, via Wikimedia Commons*

écran alphanumérique et clavier (QWERTY [73] évidemment), un Zenith [74] (figure 11). Il sert de terminal interactif et est mis à ma disposition dès le premier jour, dans mon bureau. Finis les déplacements vers un centre de calcul situé à plusieurs kilomètres pour un télétype ASR 33 ! Il y a dans l'environnement logiciel un vrai éditeur de texte plein écran [75], une révolution pour moi qui n'avais jamais utilisé qu'un éditeur ligne par ligne bogué, avec comme média le papier. Je peux voir à tout moment 15 lignes du fichier que j'édite et les modifications y apparaissent instantanément. J'ai aussi un outil d'édition de document qui permet de produire des articles qui ressemblent à ceux produits par un imprimeur [76], grâce à une imprimante laser prototype grosse comme une voiture [77]. J'y ai une adresse émail [78] et il y a là les ancêtres des réseaux

73. Piètre dactylographe, je croyais n'avoir pas d'automatismes de l'AZERTY, eh bien si ! Et donc, l'habitude de la nouvelle disposition du clavier prendra un certain temps. La démarche inverse sera un peu plus longue car, en deux ans, j'aurai plus dactylographié au MIT que pendant toute la décennie précédente.

74. Il dispose d'un écran alphanumérique sur lequel on peut afficher 80 colonnes et 24 lignes !

75. Durant cette période, j'utilise TED, un éditeur écrit en CLU. Mais je ferai quelques essais d'EMACS, le fameux éditeur de Richard Stallman écrit en LISP, que j'adopterai en rentrant en France, l'esprit de ces deux éditeurs étant le même.

76. Dire cela aujourd'hui semble incongru, puisqu'avec OpenOffice ou LaTeX cela fait partie de notre quotidien. A l'époque je n'avais jamais produit d'article dactylographié moi-même. Ainsi mes thèses ont été dactylographiées par des secrétaires de l'Institut Élie Cartan, le laboratoire de recherche de mathématiques de Nancy.

77. On trouvera sur le site [10] une photographie de la Xerox 9700. De façon facétieuse, l'imprimante du LCS (et du AI Lab) du MIT s'appelle lastminute, peut-être parce que les étudiants ont l'habitude d'imprimer leurs mémoires à la dernière minute et que l'imprimante surchargée tombe toujours en panne à ce moment là. Les pénalités infligées aux étudiants retardataires sont financières par une augmentation des frais de scolarité qui se chiffrent à plusieurs dizaines de dollars par jour.

78. Mon adresse est pierre ou pierre@mit quand je communique sur le réseau arpanet avec la côte ouest des États-Unis.

sociaux, les BBoard et la culture hacker [79] autour de Richard Stallman [80]. J'y côtoie des chercheurs de tout premier plan. Les moyens de calcul du seul LCS surpassent ceux cumulés de tous les laboratoires de recherche informatique français. Grâce à tout cet environnement, je peux me réaliser et je crée un logiciel que j'appelle REVE [81]. Grâce à lui, je produis la première démonstration automatique par ordinateur, qui est à la fois directe, subtile et fastidieuse pour un humain, d'un théorème assez surprenant de théorie des groupes, dû aux mathématiciens anglais Higman et Neumann [18]. Ce résultat a été immortalisé par le cadeau de départ que me font mes collègues (figure 12).

FIGURE 12 – La capuche offerte par l'équipe de recherche de John Guttag

Accessoirement j'apprends l'anglais et je commence à connaître le monde de la recherche internationale, notamment grâce à la doctorante Jeannette Wing [82] qui partage mon bureau, tandis que ma famille et moi nous immergeons dans la civilisation américaine. Je comprends quelques rudiments des règles du base-ball, dont John Guttag est fan, et j'améliore mon anglais en regardant la série Dynasty [83], tandis qu'à la maison, nous apprenons à nos amis américains

79. Le nom *hacker* naît au MIT et signifie *celui qui, comme un charpentier, travaille à l'herminette*. La hacker a une haute éthique et est fier de ses contributions au bien commun.

80. J'utilise quelques fois le compte rms, celui ouvert par Richard Matthew Stallman sans mot de passe, sur l'ordinateur DEC 10 du AI Lab, qui fonctionne sous le premier système d'exploitation à temps partagé ITS. Ne serait-ce que pour expérimenter ce logiciel de référence ! Stallman revendique cette absence de mot pour rendre service aux potentiels utilisateurs, ce qui est mon cas en l'occurrence.

81. Outre le sens premier du mot, il y a, dans REVE, une allusion à notre pauvre maîtrise de l'anglais, nous français qui avons tendance à mal prononcer le mot *rewriting* qui correspond à la *réécriture*.

82. Jeannette Wing deviendra, ente autres, directrice du département informatique de la NSF, à Washington. Elle est aussi l'autrice du concept de *pensée informatique* [55], (voir la traduction dans [52]). Le document a eu tellement de succès dans la communauté de l'informatique française qu'il a été traduit deux fois [51] !

83. La série (au sens télévisuel du terme) à l'intrigue facile décrit une société texane, cynique et amorale, bien différente de celle de Boston. Chaque diffusion comporte un résumé des épisodes précédents et une annonce des épisodes suivants, excellents pour l'apprentissage de l'anglais. Mais surtout les pauses publicitaires nous

à « tirer les rois », en faisant nous mêmes les galettes et en remplaçant les fèves par des dimes. La sélection hebdomadaire du Monde que je peux me procurer de temps en temps au kiosque de Harvard Square nous permet de suivre de loin en loin la politique française dont l'élection présidentielle de 1981 ; mais, de fait, nous suivons plus la politique américaine dont l'arrivée de Reagan à la Maison Blanche. Tout cela peut paraître futile, mais l'intégration dans la culture anglo-saxonne est fondamentale pour qui veut s'imprégner de la culture informatique si profondément marquée par L'Amérique [84]. Mais je reste un profond défenseur de la francisation des termes anglais. Si on a bien intégré la signification d'un concept, on utilise naturellement son équivalent français quand on parle la langue de Molière.

8 Épilogue

En revenant du MIT je retourne dans l'Ancien Monde, qui, sur le plan scientifique et par son équipement, est encore le monde d'avant, mais que je suis décidé à faire changer pour qu'il se rapproche de ce que j'ai vu aux États-Unis, et cela prendra plus d'une lustre, une décennie pour certains aspects. Je retrouve aussi un autre laboratoire de recherche à Nancy puisque Claude Pair l'a quitté pour une nouvelle carrière nationale [85].

Références

[1] Pauline Bolignano and Thierry Viéville. La découverte scientifique ... qui mérite le mérite ? *Binaire - Le Monde*, juillet 2020. https://www.lemonde.fr/blog/binaire/2020/07/.

[2] Noam Chomsky. *L'Amérique et ses nouveaux mandarins*. Seuil, 1969.

[3] Noam Chomsky. *Structures syntaxiques*. Seuil, 1969.

[4] Paul J. Cohen. On a conjecture of Littlewood and idempotent measures. *American Journal of Mathematics*, 82(2) :191–212, Apr. 1960.

[5] René Cori, Anne Michel-Pajus, and Robert Rolland. Jean-Louis Ovaert — Un homme d'action et de convictions. *Brochure IREM-APMEP*, juillet 2015. http://numerisation.irem.univ-mrs.fr/PS/IPS15006/IPS15006.pdf. URL : http://numerisation.irem.univ-mrs.fr/PS/IPS15006/IPS15006.pdf.

[6] Marion Créhange. Le compilateur Algol 60 sur IBM 1620. Claude Pair un des fondateurs de la science informatique, 14 juin 2019. https://videos.univ-lorraine.fr/index.php?act=view&id=7762.

[7] Marion Créhange and Marie-Christine Haton. L'informatique universitaire à Nancy : un demi-siècle de développement. *Technique et Science Informatiques*, 33(1-2) :127–141, 2014.

[8] Marion Créhange, Pierre Lescanne, and Alain Quéré. L'informatique de Claude Pair. *1024 – Bulletin de la société informatique de France*, 2021. À paraître.

[9] Michel Cusey and Jean-Claude Derniame. Intitiation à l'Algol à l'usage des élèves de M. P. *Institut Universitaire de Calcul Automatique*, 1968. Polycopié.

[10] Bruce Damer. The story of the Xerox 9700 electronic printing system. Digibarn Computer museum. Disponible sur Internet.

[11] J. W. de Bakker. Recursive procedures. *Mathematical Centre tracts n°24*, 1971. Mathematisch Centrum, Amsterdam.

permettent à mon épouse et à moi de discuter des points qui nous auraient échappé.

84. Par exemple, de nombreux algorithmes de files d'attente, supposent une foule disciplinée prompte à se mettre en rang, situation que l'on rencontre naturellement aux États-Unis, mais qui est si contraire à l'esprit français.

85. Claude Pair est devenu Directeur des lycées du Ministre de l'Éducation nationale, Alain Savary.

[12] Jean-Claude Derniame. À propos du cheminement dans les graphes. *1024 – Bulletin de la société informatique de France*, 16, novembre 2020 2021. disponible à https://www.societe-informatique-de-france.fr/wp-content/uploads/2020/11/1024-numero-16_Article19.pdf.

[13] Jean-Claude Derniame and Claude Pair. *Problèmes de cheminement dans les graphes*. Monographies d'informatique - AFCET. Dunod, 1971.

[14] Nachum Dershowitz. Orderings for term-rewriting systems. *Theoretical Computer Science*, 17 :279–301, 1982.

[15] Les enseignants de la Miage. MIAGE, Maîtrise de méthodes informatiques appliquées à la gestion. Plaquette de présentation, 1973-1974, 1973. Archives de l'auteur.

[16] Jean-Pierre Finance, Pierre Lescanne, Pierre Marchand, Roger Mohr, Claude Pair, Alain Quéré, and Jean-Luc Rémy. Aspects de la théorie de la programmation. Polycopié, 1974. École d'été d'Informatique – Tarbes.

[17] Monique Hecker. Un ordinateur Iris 80 à l'institut universitaire de calcul automatique. *Le Républicain lorrain*, 18 décembre 1975. Section *Meurthe-et-Moselle – Temps moderne*.

[18] G. Higman and B. H. Neumann. Groups as groupoids with one law. *Publicationes Mathematicae Debrecen*, 2 :215–227, 1952.

[19] J. Roger Hindley, Bruce Lercher, and Jonathan P. Seldin. *Introduction to Combinatory Logic*, volume 7 of *London mathematical Society, Lecture note series*. Cambridge University Press, 1972.

[20] Gérard Huet. *Résolution d'equations dans les langages d'ordre 1,2, ...,ω*. Thèse de Doctorat d'Etat, Université de Paris 7 (France), 1976.

[21] Amemiya I. and Ito T. A simple proof of the theorem of P.J. Cohen. *Bull. Amer. Math. Soc.*, pages 774–776, 1964.

[22] Donald E. Knuth. Ancient Babylonian Algorithms. *Commun. ACM*, 15(7) :671–677, 1972. doi:10.1145/361454.361514.

[23] Donald E. Knuth and P. B. Bendix. Simple word problems in universal algebras. In J. Leech, editor, *Computational Problems in Abstract Algebra*, pages 263–297. Pergamon Press, Oxford, 1970.

[24] Robert A. Kowalski. Logic for problem solving. School of Artifical Intelligence, Univ. of Edinburgh U. K., 1974.

[25] Saunders Mac Lane. *Categories for the Working Mathematician*. Springer Verlag, 1998.

[26] François le Lionnais, editor. *Les Grands Courants de la pensée mathématique*. Cahiers du Sud, 1948.

[27] Bernard Legras. *Jean Legras - Mathématicien lorrain, précurseur de l'informatique à Nancy, fondateur de l'Institut universitaire de calcul automatique*. Impr. Groupe Dialog'Guyot, Laxou, 2008.

[28] Jean Legras. *Précis d'analyse numérique*. Dunod, 1963.

[29] Jean Legras. *Initiation à l'analyse numérique*. Dunod, 1968.

[30] Damien Leloup. La France aurait-elle vraiment pu inventer Internet ? *Le Monde*, 30 mars 2021.

[31] Pierre Lescanne. *Etude de quelques théories des langages et généralisation du théorème de Kleene*. Thèse de Spécialité, Université Henri Poincaré – Nancy 1, June 1971.

[32] Pierre Lescanne. Introduction au lambda-calcul. Non publié, 1973. Disponible en ligne.

[33] Pierre Lescanne. Equivalence entre la famille des ensembles réguliers et la famille des ensembles algébriques. *RAIRO Informatique Théorique et applications*, 10(8) :57–81, 1976.

[34] Pierre Lescanne. *Etude algébrique et relationnelle des types abstraits et de leurs représentations*. Thèse de Doctorat d'Etat, Institut National Polytechnique de Lorraine, September 1979.

[35] Pierre Lescanne. Computer experiments with the REVE term rewriting systems generator. In *Proceedings of 10th ACM Symposium on Principles of Programming Languages*, pages 99–108. ACM, 1983.

[36] Pierre Lescanne. From lambda-sigma to lambda-upsilon a journey through calculi of explicit substitutions. In Hans-Juergen Boehm, Bernard Lang, and Daniel M. Yellin, editors, *Conference Record of POPL'94 : 21st ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages, Portland, Oregon, USA, January 17-21, 1994*, pages 60–69. ACM Press, 1994. doi:10.1145/174675.174707.

[37] Pierre Lescanne. *Histoire des Sciences et des Techniques*, chapter La science informatique, pages 105–116. Encyclopédie Illustrée de la Lorraine. Éditions Serpenoises, 1996.

[38] Pierre Lescanne. Les krachs sont rationnels. *Tech. Sci. Informatiques*, 32(9-10) :909–929, 2013. doi:10.3166/tsi.32.909-929.

[39] C. Livercy. *Théorie des programmes*. Dunod, Paris, 1978. Livercy est le nom d'auteur de Jean-Pierre Finance, Monique Grandbastien, Pierre Lescanne, Pierre Marchand, Roger Mohr, Alain Quéré, Jean-Luc Rémy. http://denif.ens-lyon.fr/data/programmation_enslyon/2007_sem2/biblio/Livercy.pdf.

[40] Christophe Lucas and Matteo Mio. Proof theory of Riesz spaces and modal Riesz spaces, 2021. https://arxiv.org/pdf/2004.11185.pdf. URL : https://arxiv.org/pdf/2004.11185.pdf, arXiv:2004.11185.

[41] Rex Malik. *And tomorrow ... the world ? Inside IBM*. Millington, London, 1975.

[42] Pierre-Eric Mounier-Kuhn. *L'Informatique en France, de la seconde guerre mondiale au Plan Calcul. L'émergence d'une science*. PUPS, 2010. rééd. SUP 2021.

[43] Claude Pair. *Notions sur la théorie des langages*. Université de Nancy, Faculté des Sciences, Institut universitaire de Calcul automatique, 42 avenue de la Libération, 54 Nancy, 1968. Rédigé par Alain Quéré. Archives personnelles de l'auteur.

[44] Claude Pair. CRIN. The history of a laboratory. *Annals of the History of Computing*, 12(3) :159–16, July / September 1990.

[45] Claude Pair. Colloque en l'honneur de Pierre Lescanne. Archives de l'auteur, 29 mai 2006. LORIA, Nancy.

[46] D. Plaisted. A recursively defined ordering for proving termination of term rewriting systems. Technical Report R-78-943, U. of Illinois, Dept of Computer Science, 1978.

[47] H. Gordon Rice. Recursion and iteration. *Commun. ACM*, 8(2) :114–115, 1965. doi:http://doi.acm.org/10.1145/363744.363781.

[48] Dana Scott. Lattice theoretic models for the lambda-calculus. *IFIP WG 2.2 Bulletin n°5*, 1970.

[49] Non signé. Joyau de l'informatique au service de l'université, l'Iris 80 inauguré ce jour. *L'Est républicain*, 18 décembre 1975.

[50] Karl Tombre, Long Quan, Radu Horaud, Patrick Gros, Cordelia Schmid, and Peter Sturm. In Memoriam Roger Mohr. *1024 – Bulletin de la société informatique de France*, 11 :91–98, September 2017. URL : https://hal.inria.fr/hal-01598085.

[51] Interstices (trad). Jeannette Wing. la pensée informatique. *Interstices*, 2009. Disponible sur internet.

[52] Pierre Lescanne (trad). Jeannette Wing. la pensée informatique. *Bulletin de Specif*, décembre 2008. Disponible sur internet.

[53] Peter Wegner. *Programming languages, information structures, and machine organization*. McGraw-Hill, 1968.

[54] Wikipédia. Algorithme de Rémy — wikipédia, l'encyclopédie libre, 2021. [En ligne; Page disponible le 22-avril-2021].

[55] Jeannette M. Wing. Computational thinking. *Commun. ACM*, 49(3) :33–35, 2006. URL : http://doi.acm.org/10.1145/1118178.1118215, doi:10.1145/1118178.1118215.

A Glossaire

Adresse : La mémoire d'un ordinateur est divisée en emplacements qui ont chacun une adresse.

Adressage indirect : Si une instruction identifie un emplacement de la mémoire, où chercher une information, non pas explicitement, mais par l'adresse d'un autre emplacement de la mémoire qui lui-même contient l'adresse de l'emplacement recherché, on parle d'*adressage indirect*.

Algèbre de Boole : Une structure algébrique qui sert de fondement mathématique à l'architecture des ordinateurs.

Analyse syntaxique : Opération qui consiste à examiner une chaîne de caractères pour en extraire la structure en composants, en vue d'une traduction.

Algorithme : Ordonnancement des calculs en vue de résoudre un problème. Un algorithme est mis en œuvre par un programme.

Algorithmique : Science qui étudie les algorithmes.

Arbre syntaxique (ou arbre) : Structure vers laquelle l'analyse syntaxique traduit une chaîne de caractères.

Assembleur : Programme, relativement simple, qui traduit un programme en langage d'assembleur en un programme en langage machine.

Automate : Dispositif (programme) qui traite de l'information.

AZERTY : Disposition du clavier français.

Booter : Démarrer un ordinateur en lui indiquant la première instruction qu'il doit effectuer. Sur un ordinateur d'aujourd'hui, cela est fait par une instruction à laquelle l'utilisateur lambda n'a pas accès.

Branchement : Rupture dans l'enchaînement séquentiel des instructions.

Calcul d'instruction : Un programme qui s'exécute est amené à répéter plusieurs fois la même instruction ou presque. Le calcul d'instruction est une technique absconse et obsolète qui consiste à mettre en œuvre ce « *ou presque* » et à modifier les instructions en programme au vol. Cela se fait en considérant les instructions du langage machine comme des valeurs numériques sur lesquelles on peut calculer.

Compilation : Action de traduire un programme dans un langage évolué (du tableau de l'annexe C) vers un programme du langage d'assembleur d'une machine.

Compilateur : Programme qui effectue la compilation.

Diagramme ou table de Karnaugh : Méthode graphique pour trouver ou simplifier une fonction logique à partir de sa table de vérité.

Écran bitmap : Écran qui affiche les pixels (points de base d'une image) par leurs coordonnées (horizontale et verticale) tandis que les écrans alphanumériques comme le Zenith affiche des caractères de gauche à droite, ligne à ligne. Notez que l'ASR 33 n'a pas d'écran du tout ! Les écrans bitmap (ou à tableau de bits) sont ceux de nos ordinateurs d'aujourd'hui et de nos téléphones intelligents.

Éditeur : Logiciel permettant de créer et modifier un fichier. Dans un *éditeur ligne à ligne*, les commandes s'appliquent à chaque ligne et le résultat n'est pas immédiatement visible. Dans un *éditeur plein écran*, un curseur permet de voir où se fait la modification qui est immédiatement visible.

EMACS : L'éditeur de texte de référence de l'ingénieur en informatique.

Étoile : En théorie des langages, opération sur un langage qui produit un autre langage. Par exemple l'étoile du langage $\{ab\}$, à un seul mot, s'écrit $\{ab\}^*$ et est $\{\varepsilon, ab, abab, ababab, ...\}$.

File d'attente : Structure linéaire de l'informatique dans laquelle on dispose les tâches qui attendent d'être exécutées, mise notamment en œuvre dans l'algorithme de la boulangerie.

Grammaire : Ensemble de règles décrivant mathématiquement la structure d'un langage formel. Les grammaires les plus fréquemment étudiées par les informaticiens, sont les grammaires qui définissent les *langages à contexte libre*.

Graphe : Structure faite de points reliés par des traits sur laquelle on peut exécuter de jolis algorithmes.

Instruction : La composante de base d'un programme.

Job : Tâche qui doit être exécutée par un ordinateur. En français, *tâche*.

Lambda calcul : Formalisme inventé par le logicien Alonzo Church pour abstraire la notion de calcul. Le *lambda calcul* a des liens avec la *machine de Turing*, mais il lui est légèrement antérieur et est plus mathématique.

Langage : En théorie des langages, un *langage* est un ensemble de mots ou un ensemble d'arbres. Parmi les langages, on distingue des *langages reconnaissables*, des *langages réguliers*, des *langages algébriques*.

Langage d'assembleur : Langage de très bas niveau qui représente le langage machine sous une forme lisible par un humain.

Langage machine : Langage dont les instructions sont des mots machines (donc une suite de bits) qui est fait pour être interprété directement par la machine ou le processeur. Il est presque totalement incompréhensible (directement) pour un humain.

LOAD : Instruction de chargement d'une valeur dans un registre depuis un emplacement de la mémoire.

Mot : En théorie des langages, un *mot* est un ensemble de *lettres*.

Mot clé : Dans un langage de programmation, certains mots ont un usage réservé et ne peuvent pas être utilisés comme variable, par exemple *if*, *then*, *else*, *while* etc.

Mot vide : En théorie des langages le *mot vide* est le mot qui n'a aucune lettre. Il s'écrit aujourd'hui ε. Claude Pair l'écrivait Λ, probablement parce que c'était plus facile à écrire avec une machine à écrire.

Point-fixe : Solution d'une équation de la forme $x = f(x)$.

Prix Turing : Prix décerné par l'ACM, que certains appellent le prix Nobel de l'informatique.

Précession (gyroscopique) : Particularité d'un gyroscope de se déplacer dans une direction perpendiculaire à son axe de rotation.

Programmation fonctionnelle : Façon d'aborder les programmes comme des fonctions à évaluer. Le programmeur ne donne pas l'organisation des calculs mais laisse le soin au compilateur de le faire.

QWERTY : Disposition du clavier anglo-saxon.

Récursivité : Propriété d'un programme d'être récursif.

Récursif : Une sous-programme (une procédure) est récursif (récursive), si il (elle) contient dans son code un appel à lui-même (elle-même).

Réécriture : (Pour faire simple) la *réécriture* est la théorie de la simplification des formules.

Registre : Pour accélérer le calcul, un ordinateur possède des emplacements de mémoire à accès plus rapide que le reste de la mémoire. Ces emplacements spécifiques sont appelées des registres.

Régulier : En théorie des langages, un langage est *régulier* s'il peut être engendré par des opérations simple , à savoir l'union, la concaténation et l'étoile.

Relation binaire : En mathématique une relation binaire est un concept qui englobe des l'égalité, l'équivalence où deux éléments sont associés.

Switch : Minuscule levier qui a deux positions, haut ou bas.

Système d'exploitation : Logiciel de base d'un ordinateur, qui donne accès à la machine, gère ses entrées et sorties, permet à plusieurs personnes de l'utiliser et fait fonctionner plusieurs processus en même temps. Exemples : Siris 8 (le système d'exploitation de l'IRIS 80), UNIX qui engendrera Linux, Windows, Android, MS-DOS.

Tampon : Mémoire provisoire sur laquelle sont enregistrées les données sur lesquelles l'utilisateur interagit. En anglais, *buffer*.

Temps partagé : Quand plusieurs utilisateurs ou plusieurs processus peuvent s'exécuter en même temps on parle de *temps partagé*, en anglais *time sharing*.

Traitement par lot : Quand les programmes exécutés par un ordinateur sont exécutés les uns après les autres, on parle de *traitement par lots*, en anglais *batch*. Le contraire est le temps partagé.

Terminaison : Un système de réécriture se termine si toutes les réécritures (toutes les simplifications) aboutissent à une forme « simplifiée ». La terminaison est la qualité d'un système de réécriture qui se termine.

Théorie des catégories : Théorie mathématique ayant un haut niveau d'abstraction.

Théorie des groupes : Théorie mathématique des transformations.

Triangle de Pascal : Méthode inventée par Blaise Pascal pour calculer des nombres, à savoir le nombre de combinaisons de n éléments p à p.

Unification : Résolution d'équations dans des structures mathématiques très pauvres telles que celles qu'on rencontre en informatique. Quand les structures ne contiennent que des variables qui représentent des objets de base on parle d'*unification du premier ordre*. Si, outre les objets de base, les variables représentent aussi des fonctions, on parle d'*unification du second ordre*.

UNIX : Système d'exploitation des années 1970 qui a donné lieu par la suite à Linux.

B Plan du polycopié *Initiation à l'*Algol

— Chapitre I : Généralités
— Chapitre II : Notions permettant d'écrire des programmes simples
— Chapitre III : Nombres, Variables, Expressions arithmétiques simples, Fonctions, Affectations
— Chapitre IV : Les instructions d'entrées et de sorties

— Chapitre V : ALLERA ; Instructions conditionnelles et composées
— Chapitre VI : Les ordres de bouclage : Instruction POUR
— Chapitre VII : Expression booléennes
— Chapitre VIII : Blocs et déclarations
— Chapitre IX : Les procédures
— Chapitre X : Compléments : Expression conditionnelle, Aiguillage, Expression de désignation, commentaire, La partie valeur, Instruction vide
— Annexe
— Exercices d'Algol

Parmi les exercices, j'ai noté que ce que nous appelons aujourd'hui un *tri* est appelé un *classement des éléments d'un tableau par valeurs croissantes* (Exercice 5.4).

C Quelques langages de programmation

FORTRAN	1957	Premier langage de programmation
LISP	1958	Langage des *hackers* du MIT, strictivement récursif avec une syntaxe sommaire
ALGOL 60	1960	Premier langage avec récursivité et structure de blocs
COBOL	1959	Langage pour la gestion
SYMBOL	1968	Langage d'assembleur du CII 10070
PASCAL	1970	Langage issu d'ALGOL 60, avec des structures de données et une recherche de la simplicité
CLU	1974	Langage modulaire avec types de données, récursivité gestion des exceptions, gestion de la mutabilité
OCAML	1996	Langage français essentiellement fondé sur la récursivité des fonctions et des données et sur les objets
HASKELL	$\sim$ 1995	Langage fondé sur la stricte récursivité