

CURRICULUM VITÆ DÉTAILLÉ

Pierre LESCANNE

École normale supérieure de Lyon

(2008)

<http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/tt.pdf>

Table des matières

Curriculum vitae succinct	3
Curriculum vitae thématique	8
Animation de la communauté scientifique	8
Administration de la recherche	9
Investissement pédagogique	9
Recherche	9
Diffusion des connaissances	10
Les médailles d'argent du CNRS en section 07	10
La présidence de SPECIF	11
Les cinq articles les plus significatifs	13
Description des activités d'enseignement	19
Cours enseignés actuellement	19
Cours enseignés ces dernières années	20
Diffusion des connaissances	21
Activités professionnelles spécifiques	23
Responsabilités scientifiques	23
Membre d'associations	24
Lecteur pour des revues	24
Rapporteurs de tenures, d'habilitations et de thèses	24
Responsabilités administratives	24
Liste des publications	28
Thèses	28
Livres	28
Articles	29
Communications à des conférences internationales	31
Communications à des conférences nationales	35
Descriptions de Logiciels	36
Divers et Articles Historiques	37
Documents sur WWW	38

Description détaillée des activités de recherche	39
Les Langages Formels (69-72)	39
Les Spécifications Algébriques (73-80)	40
La réécriture (80-92)	41
REVE	44
ORME	44
La déduction automatique appliquée aux protocoles (90-00) . . .	45
Le calcul avec substitutions explicites (92-...)	45
L'interprétation du calcul des séquents classique (02-...)	47
La logique épistémique et jeux (03-...)	47
Modélisation biologique (06-...)	47
Le commerce électronique (95-01)	48
EURECA	48
PLUME	49
Autres aspects de ma recherche	49

Il est recommandé de lire ce document en utilisant un lecteur de pdf qui permet de cliquer sur les liens. On le retrouve par <http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/tt.pdf>

CURRICULUM VITAE

SUCCINCT

Pierre LESCANNE né le 22 mars 1947, à Dakar Sénégal.

- 1987** L'équipe Eureca que je co-dirigeais avec Jean-Pierre Jouannaud a eu la médaille d'argent du CNRS.
- 1997-....** Professeur de classe exceptionnelle à l'école Normale Supérieure de Lyon.
- 2007-....** Vice-Président international de SPECIF, (Société des Personnels Enseignants et Chercheurs de France).
- 2007** Member of the advisory committee of the European Computer Science Summit 2007.
- 2006-....** Membre du conseil scientifique de l'ENS de Lyon.
- 2006-....** Membre du bureau d'INFORMATICS-EUROPE (association européenne des départements d'informatique universitaires pour la promotion de l'enseignement supérieur et de la recherche en informatique en Europe).
- 2005-2007** Président de SPECIF, (Société des Personnels Enseignants et Chercheurs de France).
- 2001-2004** Directeur du DEA d'Informatique Fondamentale de Lyon.
- 1997-2007** Fondateur et chef de l'équipe *Plume* du LIP de l'ENS de Lyon.
- 2001-2004** Directeur adjoint de l'école doctorale de mathématique et informatique fondamentale de Lyon.
- 1998-2001** Directeur du département de mathématiques et d'informatique de l'ENS de Lyon.
- 1997-1998** Directeur adjoint du département de mathématiques et d'informatique de l'ENS de Lyon.
- 1988-1998** Directeur de Recherche au CNRS.
- 1979-1997** Co-fondateur et chef de l'équipe-projet *EURECA* commun au CRIN et à l'INRIA-Lorraine, (partagé avec Jean-Pierre Jouannaud de 1979 à 1985)
- 1993-1996** Directeur du GDR *Programmation*.
- 1993-1998** Directeur du *Centre Charles Hermite*, Centre Lorrain de Compétence en Modélisation et Calcul à Haute Performance.

- 1989-....** Éditeur du journal *Applicable Algebra in Engineering, Communication and Computing* publié par Springer Verlag.
- 1998-....** Membre du IFIP W.G. 1.3 et W.G. 1.6
- 1986-1992** Directeur adjoint du Centre de Recherche en Informatique de Nancy.
- 1983-1984** Faisant fonction de Professeur à l'Université de Nancy 2.
- 1980-1988** Chargé de Recherche au CNRS.
- 1980-1982** Visiting scientist au Laboratory for Computer Science au MIT.
- Sep. 1979** Docteur d'État.
- 1974-1980** Attaché de Recherche Agrégé au CNRS.
- 1972-1974** Maître-Assistant à l'Université de Nancy 2.
- 1971-1972** Sous les drapeaux.
- 1968-1971** Assistant puis assistant agrégé à l'Université de Nancy 1.
- 1970** Champion de l'Académie de Nancy-Metz de rugby avec l'équipe de la Faculté des Sciences.
- 1969** Reçu à l'agrégation de mathématiques, 49^{ème} sur 137.
- 1964-1968** Étudiant à la Faculté des Sciences de Nancy.

Domaines de recherche

Méthodes formelles, informatique fondamentale et informatique théorique, logique, théorie des types et leurs applications, logique de la connaissance, théorie des jeux appliquée à la biologie.

Thèmes de recherche

Actuels

- Lambda-calcul, substitutions explicites, théorie des types.
- Dédution automatique d'ordre supérieur et ses applications (notamment à la théorie des jeux)
- Programmation fonctionnelle et modèles de calcul.
- Logique épistémique, logique de la connaissance et rationalité.
- Théorie des jeux non coopératifs et application à la biologie)
- Application de la théorie des jeux à la génomique.

Passés

- Dédution automatique et réécriture, création des logiciels REVE et ORME.
- Spécifications algébriques et langages formels.

Publications

- coauteur d'un livre
- 23 articles dans des journaux internationaux, dont 10 comme seul auteur.
- 10 fois éditeur de livres ou numéros spéciaux de revues
- 35 communications à des conférences internationales
- 22 communications à des conférences nationales et réunions de travail
- 2 logiciels REVE et ORME
- Membre de la rédaction du journal *Applicable Algebra in Engineering, Communication and Computation*, publié par Springer Verlag.

Thèses d'Université dirigées et soutenues

- Fernand Reinig** *L'Ordre de Décomposition : un Outil Incrémental pour Prouver la Terminaison Finie des Systèmes de Réécriture équationnels* Thèse de troisième cycle (1981) partiellement encadré ¹ Administrateur-Directeur du Centre de recherche public – Centre universitaire (Crp-Cu) du Luxembourg.
- Isabelle Gnaedig** : *Preuves de terminaison des systèmes de réécriture associatifs commutatifs : une méthode fondée sur la réécriture elle-même*, Thèse de troisième cycle. (1986). Chargée de recherche à l'INRIA
- Ahlem Ben Cherifa** : *Preuves de terminaison de systèmes de réécriture basées sur des calculs élémentaires sur des polynômes*. Thèse de l'Université Nancy I (1986). Maître de conférences à l'Université de Tunis.
- Azzedine Lazrek** : *Étude et réalisation de méthodes de preuve par récurrence en logique équationnelle*. Thèse de l'Institut National Polytechnique de Lorraine (1988). Maître de conférences à l'Université de Marakkech.
- Jocelyne Rouyer** : *Calcul formel en géométrie algébrique réelle appliqué à la terminaison des systèmes de réécriture*, Thèse de l'Université Nancy I (1991). Maître de conférences retraitée de l'Université Henri Poincaré.
- Joseph Rouyer** : *Développement d'algorithmes dans le calcul des constructions*. Thèse de l'Institut National Polytechnique de Lorraine (1994). Professeur agrégé à l'Université Henri Poincaré.
- Stefan Krischer** : *Méthodes de vérification de circuits digitaux*. Thèse de l'Institut National Polytechnique de Lorraine (1994). Collaborateur à l'Office européen des Brevets à Munich.
- Boutheina Chetali** *Vérification formelle des systèmes parallèles décrits en UNITY à l'aide d'un outil de démonstration automatique* Thèse de l'Université Henri Poincaré Nancy I (1996), CC and Formal Methods Manager, chez AXALTO (leader mondial dans la production de cartes à puces).
- Zine-el-Abidine Benaïssa** *Les calculs de substitution explicites comme fondements des langages de programmation fonctionnels*. Thèse de l'Université Henri Poincaré Nancy I (1997). Responsable d'équipe au laboratoire de

¹Je suis parti pour deux ans aux États-Unis en 1980 et Jean-Pierre Jouannaud a repris l'encadrement.

recherche et développement d'INTEL détaché chez Microsoft à Seattle (Washington, États-Unis)

Briaud, Daniel *Substitutions explicites et unification d'ordre supérieur* (1997)
Thèse de l'Université Henri Poincaré Nancy I.

Barbara Heyd *Application de la théorie des types et du démonstrateur COQ à la vérification de programmes parallèles*. Thèse de l'Université Henri Poincaré Nancy I (1997). Ingénieur, chef de projet en entreprise.

Frédéric Lang *Modèles de la β -réduction pour les implantations*. Thèse de l'ENS de Lyon (1998). Chargé de recherche à l'INRIA Rhône-Alpes

Frédéric Prost *Interprétation de l'analyse statique en théorie des types*. (1999)
encadré lors de sa dernière année de thèse. Maître de Conférences à l'Université Joseph Fourier de Grenoble.

Silvia Likavec *Types for functional and object-oriented languages* (2005), thèse en cotutelle avec l'Université de Turin.

Romain Kervarc *Les substitutions explicites dans le cube de Barendregt et dans les séquents* (2007). Ingénieur de recherche à l'ONERA.

Dragiša Žunić *Computing with sequents and diagrams in classical logic - calculi $*X$, $@X$ and dX* . (2007)

Stéphane Le Roux *Généralisation et formalisation en théorie des jeux. des jeux* (2008) (post-doc dans le projet commun l'INRIA Futurs et Microsoft).

Habilitations et thèse d'État comme promoteur

Françoise Bellegarde : *Utilisation des systèmes de réécriture d'expressions fonctionnelles comme outil de transformation de programmes itératifs*,
Thèse d'état (1985). Professeur à l'Université de Franche Comté à Besançon.

Roberto Amadio *Trois Lambda-Calculs*. Professeur à l'Université de l'Université Paris 7.

Nicolas Hermann *Divergence des systèmes de réécriture et schématisation des ensembles infinis de termes*. (1984). Chargé de recherche au CNRS.

Séjours scientifiques

mars 2004, mars 2005, mars 2006, mars 2007 Visiteur invité au Japan Advanced Institute of Science and Technology, invité par les professeurs Futatsugi and Vestergaard.

juillet 1987 Consultant scientifique à l'Institute for New Generation Computer Technology à Tokyo au Japon.

juillet-août 1985 Visiteur Scientifique au Computer Science Laboratory du SRI-International, invité par Joseph Goguen.

juillet-août 1984 Visiteur Scientifique au Laboratory for Computer Science du MIT, invité par John Guttag.

Membre d'instances

1993-1999 Membre nommé du Conseil National des Universités.

1992-1994 Vice Président de SPECIF, (Société des Personnels Enseignants et Chercheurs de France).

1983-1987 Membre du Comité National de la Recherche Scientifique, section 07.

1983-1987 Membre du Conseil de Département des Sciences Pour l'Ingénieur du CNRS.

1999-2002 Membre du conseil d'administration de l'ENS de Lyon.

2003-.... et 1989-1994 Membre du conseil d'administration de SPECIF

2006-.... Membre du comité de pilotage du programme sécurité de l'Agence nationale de la recherche.

2006-.... Membre du conseil scientifique de l'ENS de Lyon.

2006-.... Membre du bureau d'INFORMATICS-EUROPE.

CURRICULUM VITÆ

THÉMATIQUE

Je vais adopter ici une structuration plus thématique que chronologique, préférant une mise en évidence de faits plutôt qu'une liste exhaustive.

Animation de la communauté scientifique

Au cours de ma carrière j'ai consacré une grande partie de mon temps à l'animation de la recherche.

Direction des équipes de recherche Castor, Euréca et Plume

Je suis responsable d'une équipe de recherche depuis 1978, de l'équipe Castor avec Jean-Pierre Finance en 1978-1980, de l'équipe Euréca, en coopération avec Jean-Pierre Jouannaud de 1982 à 1987, puis seul de 1987 à 1997. Depuis 1997 et mon arrivée à l'ENS de Lyon à 2006, j'ai dirigé l'équipe Plume du LIP.

Participation à des comités éditoriaux

Journal *Applicable Algebra* dont je suis membre du comité éditorial depuis 1990.

Conférences j'ai été membre de seize comités de conférences internationales (voir liste page 23) dont entre autres, LICS et RTA (fondateur et cinq fois membres)

Numéros spéciaux de journaux j'ai édité des numéros spéciaux de journaux dans *Theoretical Computer Science* (vol. 67 (2 & 3)), *Journal of Logic Programming* (vol. 12(3 & 4)), *Applicable Algebra in Engineering, Communication and Computing* (vol 12(1 & 2),) , *Lectures Notes in Computer Science* (vol. 256, 343 et 501) et *Electronic Notes in Theoretical Computer Science* (vol. 140).

Membre de groupes IFIP

Je suis membre du groupe IFIP 1.6 *Term Rewriting* et membre émérite du groupe IFIP 1.3 *Foundations of System Specification*. Je suis aussi modérateur de la liste *rewriting* que j'ai créée en 1987 (238 abonnés).

Administration de la recherche²

Directeur du GDR *Programmation* de 1992 à 1996.

Membre du comité national du CNRS ainsi que du comité de Département des Sciences Physiques pour l'Ingénieur de 1982 à 1985.

Membre du CNU de 1993 à 1999.

Présidence de SPECIF Je suis actuellement vice-président de SPECIF après avoir été président en 2005-2006 et j'ai été vice-président recherche en 1992-1994. Outre les activités habituelles, j'y ai initié une coopération internationale qui à l'heure de l'Europe me paraît primordiale.

Directeur adjoint du CRIN, de 1986 à 1992 (le CRIN groupait 12 équipes de recherche, 200 enseignants-chercheurs et chercheurs, dont 20 chercheurs CNRS et 20 chercheurs INRIA).

Directeur du Centre Charles Hermite, centre lorrain de compétence en calcul à hautes performances, de 1993 à 1998.

Investissement pédagogique

J'ai enseigné dans des contextes aussi différents que le DEUG de psychologie à l'Université de Nancy II, que le CNAM en cours du soir, que le département informatique de l'IUT de Nancy, que la MIAGE de Nancy (que j'ai contribué à fonder), que l'École normale supérieure de Lyon, que l'École des Mines de Nancy ou que l'Université Claude Bernard.

J'ai participé de façon déterminante au premier livre, non seulement en France, mais dans le monde, sur la théorie des programmes et la sémantique : le «*Livercy*».

Administration de l'enseignement

Direction du département de Mathématique et Informatique (1998-2001) (une journée par semaine).

Directeur du DEA d'informatique et directeur adjoint de l'école doctorale Math-Info (2001-2004) (une demi-journée par semaine).

Recherche

Tout au long de ma carrière, j'ai été un chercheur actif et je le reste. Les débuts ont été difficiles, car dans les années soixante-dix, il n'était pas simple d'être un chercheur en informatique, nous n'avions pas de matériel adéquat (mon choc a été grand quand je suis arrivé aux États-Unis en 1980), la recherche en informatique théorique et fondamentale n'avait pas l'organisation et le statut qu'elle a maintenant, je ne maîtrisais pas l'anglais. Bref, il fallait tout inventer et tout créer. Je suis très fier que le CRIN, devenu le LORIA,

²Toutes ces activités correspondent grosso modo à un mois et demi de travail par an ou une journée par semaine.

soit ce qu'il est aujourd'hui. A la suite de Claude Pair, j'ai d'abord travaillé sur les *spécifications*. Dès cet époque (1974) j'ai programmé mes algorithmes (en Pascal), notamment un petit système de preuves équationnelles, dans des conditions héroïques : des paquets de cartes, dans un centre de calcul isolé du monde, j'étais le seul à Nancy, à l'époque, à programmer du calcul symbolique. On m'associe souvent à la *réécriture*, c'est d'une certaine manière un cliché, mais c'est aussi vrai qu'avec Gérard Huet, j'en suis l'un des prometteurs en France, ayant initié en particulier Jean-Pierre Jouannaud et beaucoup d'autres au domaine. J'ai aussi créé, REVE, le premier logiciel sur la réécriture. Aujourd'hui ma recherche se fait dans le cadre du LIP et porte sur la formalisation de la *logique de la connaissance* et sur les *modèles computationnels de la logique classique*.

Diffusion des connaissances

La diffusion des connaissances à un large public me paraît un devoir essentiel d'un enseignant du supérieur. Dans ce cadre, j'écris des articles dans la revue *La Recherche* [D-39, D-42] (voir page 38), je participe à la *Semaine de la Science* et je rédige des articles pour Wikipédia.

La médaille d'argent du CNRS 86-87

La définition de la *médaille d'argent* du CNRS est la suivante :

La Médaille d'argent du CNRS distingue des chercheurs, pendant leur ascension, mais déjà reconnus sur le plan national et international pour l'originalité, la qualité et l'importance de leurs travaux.

Voici la liste des médailles d'argent du CNRS telles qu'elles ont été attribuées en section 07 depuis 1985 telle que j'ai pu les reconstituer. On notera que certaines années, les médailles d'argent, qui sont attribuées par département, n'ont pas été attribuées à la section 07.

- 2007** : *Jean-Luc Gauvain*,
- 2006** : pas de médaille en section 07,
- 2005** : *Jacques Stern*,
- 2004** : *Philippe Flajolet*,
- 2003** : *Philippe Cinquin*,
- 2002** : pas de médaille en section 07,
- 2001** : *Joseph Sifakis*,
- 2000** : pas de médaille en section 07,
- 1999** : *Patrick Cousot*,
- 1998** : pas de médaille en section 07,
- 1997** : pas de médaille en section 07,
- 1996** : *Marie Claude Gaudel*,

1995 : pas de médaille en section 07,
1994 : pas de médaille en section 07,
1993 : *Jean-Claude Laprie*
1992 : *Xavier Viennot*
1991 : pas de médaille en section 07,
1988-1990 information non disponible
1986-1987 : *Équipe Euréca du CRIN*
1985 : *Michel Las Vergnas*

L'attribution de la médaille d'argent du CNRS en 1986-1987 à l'équipe EURECA que je codirigeais a été assortie de la citation qui suit.

L'équipe (EURECA) «Étude de la réécriture et de ses applications» fut créée début 1981 par Jean-Pierre Jouannaud jeune professeur à l'université Nancy 2 et Pierre Lescanne chargé de recherche au CNRS. Parmi ses membres on doit ajouter Jean-Luc Rémy, Claude Kirchner, Hélène Kirchner, chargés de recherche au CNRS. L'équipe EURECA comprend en outre Françoise Bellegarde, Ahlem Ben Cherifa, Isabelle Gnaedig, Emmanuel Kounalis, Miguel Muñoz, Jalel Mzali, Fernand Reinig, René Schott, Michael Rusinowitch et Hantao Zhang.

EURECA est devenue rapidement l'une des équipes phares au niveau international, dans le domaine de la réécriture et de la preuve équationnelle. Ses résultats les plus remarquables concernent d'une part, l'étude des mécanismes fondamentaux de la réécriture (termination des systèmes de réécriture, unification dans les théories équationnelles), d'autre part les généralisations de la réécriture (réécriture équationnelle, réécriture conditionnelle, réécriture dans les algèbres ordonnées, utilisation de la complétion pour les preuves par récurrence, superréduction).

Ces avancées théoriques ont systématiquement donné lieu à des réalisations logicielles :

- *REVE, logiciel de preuves par réécriture réalisé en collaboration avec le MIT et plus récemment avec l'université de Stony Brook,*
- *OBJ2, un langage de spécification algébrique, réalisé en collaboration avec le SRI.*

Ces réalisations ont été un vecteur essentiel du renom international d'EURECA. Elles ont valu à ses membres de nombreuses invitations à l'étranger. EURECA a porté très haut et très loin les couleurs de l'informatique nancéienne et de son laboratoire le CRIN.

La présidence de SPECIF

Extrait du bulletin SPECIF n°57, page 92 :

«Conformément à ce qu'il avait annoncé, Pierre Lescanne a quitté ses fonctions de président, qu'il a assurées pendant deux années qui furent riches et actives. Le CA lui exprime ici ses remerciements pour le travail accompli. L'image et le rayonnement de SPECIF ont largement bénéficié du dynamisme et de l'enthousiasme que Pierre Lescanne a mis au service de la communauté des informaticiens.»

MY FIVE MOST SIGNIFICATIVE PAPERS

Cette partie qui m'avait été demandée pour un autre dossier destiné à être expertisé internationalement a été écrite en anglais et j'ai décidé de le laisser telle quelle. Comme ce qui fait un chercheur n'est pas le nombre de ses publications, mais leur valeur, je me suis livré à l'exercice consistant à mettre en évidence mes cinq meilleures publications, car c'est la seule façon d'attester de l'impact et de la qualité de la recherche d'un scientifique.

Let me first recall few facts. I defended my first thesis (thèse de troisième cycle) in 1971 and I defended my Thèse d'Etat at the University of Nancy in 1979 both under the supervision of Professor Claude Pair. After taking teaching positions at the University of Nancy, I entered CNRS in 1974, which I left in 1997 to join Ecole normale supérieure de Lyon. During the years 1980-1982, I stayed at MIT in professor John Guttag group. My main contribution during the period 1970-1980 was the textbook *Theory of Programs* [Liv78]³ (in French) with co-authors, my main contribution was on fixed-point, verification and denotational semantics including continuations. In *Mathematical Reviews* Jean Vuillemin wrote

This is an excellent book on the "Theory of programming". The author takes us, in a unified manner, through the various theories and lessons of the field. He introduces fixed-points and program schemes; from there we get to verification and construction of programs. The author then puts all this together, and constructs a programming language, together with various expressions of its semantics. This book could serve as the basis for a graduate course on the subject.

Below I present my most significant papers⁴ also the most cited papers of CiteSeer <http://citeseer.ist.psu.edu/allcited.html>. The papers are on the Web at <http://perso.ens-lyon.fr/pierre.lescanne/5MostSignificative.htm>.

The presentation will be subdivided in three sections according to three main topics : *computer environments for rewriting, disequations* and *explicit*

³Bibliographic references are given at the end of this chapter.

⁴☺ I am talking about my scientific papers, since the note that made me known among the community of French young scientists is probably my Web page "How to apply?" [Les90a] which tells young researchers how to prepare an application for an academic position.

substitutions.

It should be noticed that in computer science some conferences have an impact equivalent to the best journals [Pat04]. This is the case of the *ACM Conference on Principles of Programming Languages* nicknamed POPL. This is also why two such communications at POPL appear in my top 5.

Computer environments for rewriting

In my career as a computer scientist I have worked on foundation of computer science, but I have also implemented algorithms in software which have been reused by other persons. This was the case of REVE an environment for proving property of term rewriting systems that I developed at MIT in 1980 and 1981 in the group of Professor John Guttag.

Many mathematical theories are presented by identities (or equalities) as are also theories associated with some abstract data types, which are tools used in the design of software. To prove that an equality is a consequence of old ones, the best way is to orient the given equalities into simplifying rules (called rewrite rules) and to try to simplify the sides of the equality to be proved till one gets a trivial equality. But usually orienting the given equalities into rules is not easy. First, one has to be sure that the simplifying process terminates in all case, avoiding divergence and one has to be sure that the most simplified form of an expression is unique. Donald Knuth [KB70] has designed a procedure that starts from a set of equalities and provides a set of rules which is equivalent to the set of equalities, which terminates always and which provides always a unique most simplified expression. The so called Knuth-Bendix procedure relies on a method for proving that a set of rules terminates. Actually there is no universal method for proving that a set of rules terminates, so several methods exist with each a specific domain of application.

Computer Experiments with the REVE Term Rewriting System Generator *10th ACM Symposium on Principles of Programming Languages*, January 24-26 1983, Austin Texas, pp 99-108. In this paper, two appendices propose the full development of two examples : an equational specification of an abstract data types for sets and an automatic proof of an inductive lemma about the Fibonacci numbers. The main originality of REVE is its tool for automatically proving the *uniform termination* (also known as noetherianity or strong normalization) of the rewriting systems being built. The method based on the *recursive decomposition ordering* [Les90c] is *incremental*, which means REVE builds the proof of termination as it discovers new rules. The *status* of an operator (introduced for the first time) tells how the ordering should consider it and is also incrementally introduced. In this paper, I announce a very important result, namely the first proof done by REVE of a non trivial mathematical result by a computer with no help (or almost no help) from a human. This result says that the presentation by the unique equation

$$x/(((x/x)/y)/z)/(((x/x)/x)/z) = y$$

proposed by Higman and Neuman [HN52] is actually this of group. In [Les84] I present a full description of the proof with other results obtained in algebras. Whereas the proof of Higman and Neumann uses second order argument i.e., makes quantifications over sets, REVE uses first order.

REVE was later developed by the MIT group and gave birth to a new tool called the LARCH PROVER which was used to handle huge proofs related to the foundation of computer science, for instance a practical concurrent garbage collector, a very sophisticated, subtle and error prone piece of software. Later I wrote a new software called ORME [Les90b], based on a rigorous approach mostly for a didactic purpose.

Termination of Rewriting Systems by Polynomial Interpretations and its Implementation *Science of Computer Programming, vol. 9, n° 2, (1987), pp 137-160.* This work has been done with a Tunisian student, named Ahlem Ben Cherifa who works now in a business company in her country⁵. The paper describes an automatic implementation of a method for proving termination of rewriting systems based on polynomial interpretations. The key point is to fully automatically prove that a polynomial is positive for every value. The proposed heuristics is simple and works well. It is the basis of many tools developed these days. Since I extended this method to handle elementary interpretations (polynomials + exponentials) [Les95] and I proposed a method based on transformation of rewrite systems [BL90].

Disequations

Solving problems containing only formal equations is fundamental in symbolic computation, in computer aided theorem proving and in constraint solving. It is also an important theoretical issue.

Equational problems and disunification *J. of Symbolic Computation (1989), vol. 3 and 4, pp. 371-426.* This paper is a collaboration with my colleague Hubert Comon who is now professor at the ENS de Cachan. In equational theories and in rewriting, one has often to solve equations. Working with one of my first master students Jean-Jacques Thiel [Thi84, LLT90], we saw that one has also to solve disequations. Whereas equations are questions of the form $s \stackrel{?}{=} t$, *disequations*, a terminology that Hubert and I made popular, are questions of the form $s \stackrel{?}{\neq} t$. In this paper we proposed a new algorithm for solving equational problems, i.e., problems where the only predicate is $=$, this means they contain equations, disequations and quantifications. The algorithm is based on inference rules and we show that those rules form a complete procedure for finding solutions of equational problems if they exist. As a by product we give a algorithm for deciding the validity in the Herbrand universe of first order formulae that contain only $=$.

⁵Her first name Ahlem means *dream* (alias *rêve* in French).

Explicit substitutions

λ -calculus was designed to formalize the concept of function. This description is intentional, that is that roughly speaking it tells how a function yields a result when applied to a value. It relies on the concept of substitutions which is defined in the epi-theory of the λ -calculus ($\epsilon\pi\iota$ is “around” in Greek).

From lambda-sigma to lambda-epsilon, a journey through calculi of explicit substitutions *21st ACM Symposium on Principles of Programming Languages (POPL), 16-19 January 1994, Portland, Oregon, pp 60-69.* The fact that a main concept namely substitution was not defined in the λ -calculus stroke me. In addition, with my first-order culture, I disliked the fact that the λ -calculus had to speak about bound variables, capture avoiding and like (since I partly changed my mind). At the beginning of the nineties, researchers proposed the concept of “explicit substitutions” which with de Bruijn indices *internalizes* the substitution and moreover makes the calculus *a true first order calculus*. But I found this proposal too complex and oriented toward one approach. Therefore I suggested many possible systems and eventually one named $\lambda\nu$ or *lambda-epsilon*. $\lambda\nu$ is extremely simple as it is made of eight tiny and straightforward rewrite rules. This paper with a rather provocative title contributed to open a new active and fecund field which sets a new foundation of λ -calculus and logic and which applies to the understanding of functional programming language implementation. Its impact was both among logicians and among specialists in semantics of programming languages.

Lambda-epsilon, a calculus of explicit substitutions which preserves strong normalisation *J of Functional Programming, Vol. 6, n° 5 (September 1996).* This paper was written with my Nancy PhD students Zine-El-Abidine Benaissa, and Daniel Briaud and my Nancy colleague Jocelyne Rouyer. In 1994 a question was raised namely whether calculi of explicit substitution represent faithfully λ -calculus. Among others, people wanted to know whether strong normalization was preserved, where *strong normalization* of a term is the property that says that all reductions starting at that term terminates. In 1995 two main results were announced

1. The calculus of explicit substitution $\lambda\mathbf{x}$ (due to Roel Bloo and Kristofer Rose [BR95], see also [DL03, LLD⁺04]) and the calculus of explicit substitution $\lambda\nu$ preserve strong normalization.
2. The calculi of explicit substitution $\lambda\sigma$ and $\lambda\sigma_{\uparrow}$ do not preserve strong normalization (Mellies [Mel95]).

Therefore, people realize that *preservation of strong normalization* is an important property of calculi of explicit substitution which is not shared by all other formalisms. This started many researches for new calculi which enjoy preservation of strong normalization and more properties in order to ensure that the λ -calculus is faithfully extended of as mentioned above. The method proposed in the paper was the source of many others.

Bibliography on the most significant publications

- [AC98] R. M. Amadio and P.-L. Curien. *Domains and Lambda-calculi*. Cambridge University Press, Cambridge, 1998.
- [Bar84] H. P. Barendregt. *The Lambda-Calculus, its syntax and semantics*. Studies in Logic and the Foundation of Mathematics. Elsevier Science Publishers B. V. (North-Holland), Amsterdam, 1984. Second edition.
- [BD02] M. Broy and E. Denert, editors. *Software Pioneers*. Springer-Verlag, 2002.
- [BL90] F. Bellegarde and P. Lescanne. Termination by completion. *Applicable Algebra in Engineering, Communication and Computation*, 1(2) :79–96, 1990.
- [BR95] Roel Bloo and Kristoffer Høgsbro Rose. Preservation of strong normalisation in named lambda calculi with explicit substitution and garbage collection. In *CSN '95 – Computer Science in the Netherlands*, pages 62–72, November 1995.
- [DL03] Daniel Dougherty and Pierre Lescanne. Reductions, intersection types, and explicit substitutions. *Mathematical Structures in Computer Science*, 13(1) :55–85, 2003.
- [HN52] G. Higman and B. H. Neumann. Groups as groupoids with one law. *Publicationes Mathematicae Debrecen*, 2 :215–227, 1952.
- [KB70] 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.
- [Les84] P. Lescanne. Term rewriting systems and algebra. In R. Shostak, editor, *Proceedings 7th International Conference on Automated Deduction, Napa Valley (Calif., USA)*, LNCS. Springer-Verlag, 1984.
- [Les90a] P. Lescanne. Comment candidater ? <http://perso.ens-lyon.fr/pierre.lescanne/TEXTS/candidater.html>, 1990.
- [Les90b] P. Lescanne. Implementation of completion by transition rules + control : ORME. In Hélène Kirchner and W. Wechler, editors, *Proceedings 2nd International Conference on Algebraic and Logic Programming, Nancy (France)*, volume 463 of LNCS, pages 262–269. Springer-Verlag, 1990.

- [Les90c] P. Lescanne. On the recursive decomposition ordering with lexicographical status and other related orderings. *Journal of Automated Reasoning*, 6 :39–49, 1990.
- [Les95] P. Lescanne. Elementary interpretations in proofs of termination. *Formal Aspect of Computing*, 7 :77–90, 1995.
- [Liv78] C. Livercy. *Théorie des programmes*. Dunod, Paris, 1978. Livercy stands for Jean-Pierre Finance and Monique Grandbastien and Pierre Lescanne and Pierre Marchand and Roger Mohr and Alain Quéré and Jean-Luc Rémy. Available at <http://perso.ens-lyon.fr/pierre.lescanne/publications.html>.
- [LLD⁺04] Stéphane Lengrand, Pierre Lescanne, Dan Dougherty, Mariangiola Dezani-Ciancaglini, and Steffen van Bakel. Intersection types for explicit substitutions. *Information and Computation*, 189(1) :17–42, 2004.
- [LLT90] A. Lazrek, P. Lescanne, and J.-J. Thiel. Tools for proving inductive equalities, relative completeness and ω -completeness. *Information and Computation*, 84(1) :47–70, January 1990.
- [Mel95] P.-A. Melliès. Typed λ -calculi with explicit substitution may not terminate. In M. Dezani and G. Plotkin, editors, *TLCA '95*, volume 902 of *LNCS*, pages 328–334. Springer-Verlag, 1995.
- [Pat04] David A. Patterson. The health of research conferences and the dearth of big idea papers. *Commun. ACM*, 47(12) :23–24, 2004.
- [Thi84] J.-J. Thiel. Stop losing sleep over incomplete data type specifications. In *Proceeding 11th ACM Symp. on Principles of Programming Languages*, pages 76–82. ACM, 1984.

DESCRIPTION DES ACTIVITÉS D'ENSEIGNEMENT

J'ai commencé ma carrière par un poste d'assistant (68-71) puis de maître assistant (72-74), entrecoupé d'un service militaire comme professeur à l'école de maistrance de l'aéronautique navale, au cours duquel j'ai d'ailleurs suivi le seul stage de formation pédagogique de ma carrière. Durant cette période j'ai enseigné les mathématiques à des étudiants en DEUG de physique, en DEUG de psychologie et l'IUT département informatique. J'ai participé à la création de la MIAGE de Nancy (l'une des trois premières MIAGE de France pour laquelle j'ai contribué en pionnier à l'élaboration des programmes) et j'ai fait des cours du soir au CNAM. J'ai de plus fait un échange de fonction avec Jean-Pierre Jouannaud durant l'année universitaire 1983-1984 où j'ai exercé les fonctions de professeur à l'Université de Nancy 2 à la MIAGE notamment. J'ai enseigné chaque année dans le DEA d'informatique de Nancy. De plus, en 1989-90 j'ai enseigné la programmation fonctionnelle et ML à l'école des Mines de Nancy, malheureusement ce cours qui avait plu aux élèves et à leur enseignant n'a pas pu être reconduit. J'ai participé activement (rédigeant les parties les plus innovantes et les plus difficiles) à l'écriture du premier livre français d'informatique théorique [L-1], aussi l'un des premiers sur le plan international et j'ai traduit le livre de Liskov et Guttag [TR-1]. En temps que professeur à l'ENS de Lyon j'assure trois cours de 16 séances chacun. J'utilise les nouvelles technologies de communication, en particulier mes cours sont fondés sur la vidéoprojection (pour plus d'information se reporter à ma page WEB <http://perso.ens-lyon.fr/pierre.lescanne/enseignement.html>).

Cours enseignés actuellement

Ces cours font partie de la licence d'informatique fondamentale de Lyon et du master recherche d'informatique fondamentale de Lyon.

Cours de logique, ce cours se situe au premier semestre (première année d'ENS, troisième année de licence) et aborde la logique propositionnelle intuitionniste et classique sous tous leurs aspects : à la Hilbert, en déduction naturelle et en calcul des séquents. J'y insiste sur la seconde, eu égard à ses applications en

informatique. La correspondance de Curry Howard situe les liens entre calculs et preuves ; il me permet d'y faire une introduction au lambda-calcul développée sur plusieurs séances. Le calcul des prédicats donne les bases du raisonnement et sa meilleure application dans un cours de logique est l'axiomatique de la théorie des ensembles de Zermelo-Fraenkel. La logique de la connaissance sur la base du livre de Fagin, Halpern, Moses et Vardi ouvre un champ nouveau à la logique. Le public est composé d'étudiants en mathématiques, d'étudiants en informatique et d'un étudiant en philosophie ; je peux constater que la culture des taupins est très éloignée de ce dont je leur parle et que des explications détaillées, parfois philosophiques et historiques répondent à des interrogations que se posent les étudiants, par exemple sur ce qu'est une preuve du point de vue formel ou sur la différence «valide-prouvable?», autrement dit sur la notion «de vrai en mathématiques et en informatique».

Cours de réécriture, situé d'abord en première année d'ENS, puis maintenant en deuxième année (première année de master) j'y couvre la réécriture du premier ordre en suivant le livre de Baader et Nipkow (le *All that*), mais aussi en m'en démarquant, notamment en introduisant l'unification d'ordre supérieur.

Cours de sémantique, deuxième année (première année de master) j'y aborde la sémantique formelle des langages de programmation : sémantique opérationnelle (réductions petites étapes et grandes étapes, machines abstraites, continuations, graphes d'interaction) sémantique dénotationnelle (cpo, modèles du lambda calcul), sémantique axiomatique (pragmatique, correction et complétude relative), sémantique des langages à objets, compléments de théorie des types, types avec intersections, système F.

Histoire des sciences : *histoire des algorithmes*, dans le cadre de ce cours (2002-2003, 2003-2004, 2004-2005) qui s'adresse surtout aux étudiants du master d'*histoire, philosophie et didactique des sciences* (autrefois dans le DEA Construction des Savoirs Scientifiques de l'école doctorale Informatique et Information pour la Société de Lyon), Pierre Crépel m'a demandé d'intervenir sur l'histoire des algorithmes. Je me suis focalisé sur trois dates-clés : les algorithmes chez les Babyloniens du temps d'Hammourabi et sous la période Séleucide, l'algorithme d'Euclide dans ses fameux livres, enfin la formalisation de la notion d'algorithme à Princeton au début des années 1940.

Logique épistémique, ce cours (une séance) a été fait en novembre 2004 à l'invitation d'Alain Mille dans le cadre de la filière *Connaissance et Cognition* du master recherche en informatique de Lyon.

Cours enseignés ces dernières années

Module «Fondements» du Magistère MIM, ce cours situé au premier trimestre de la première année, n'a véritablement fonctionné qu'une année.

J'ai introduit, dans une vision informatique et en constante référence à celle-ci, les différents modes de raisonnement par récurrence (classique, généralisée, noethérienne, structurelle et sur les multi-ensembles) le lemme de Koenig et les théorèmes du point-fixe. Ensuite j'ai abordé les systèmes formels du point de vue la logique équationnelle, puis j'ai introduit le lambda-calcul et ses théorèmes de base.

Cours de recherche «Terminaison des systèmes de réécriture» du DIL, ce cours avait pour but d'introduire aux grands théorèmes du domaine, le théorème de Kruskal et le théorème de forte normalisation du système F. Ayant du rappeler les principaux concepts du domaine, je n'ai pu traiter que le premier.

Cours de base du DEA d'informatique de Lyon sur «la logique et la logique de la connaissance», il s'agissait d'un cours assez difficile à assurer en raison du niveau hétérogène des étudiants, certains souhaitent des rappels tandis que d'autres demandent des connaissances vraiment nouvelles par rapport à l'enseignement de maîtrise qu'ils ont reçu. J'ai décidé d'orienter mon cours autour de la logique épistémique (dite aussi logique de la connaissance), je me suis inspiré largement du livre de Fagin, Halpern, Moses et Vardi *Reasoning about knowledge*. J'ai utilisé le début du cours pour faire des rappels sur la logique (théorie de la preuve et théorie des modèles, complétude et correction, validité et prouvabilité). La fin du cours a été consacré à la logique épistémique et à ses applications aux systèmes distribués et multi-agents.

Travaux dirigés de complexité avancé, en 2000 je me suis lancé un défi qui consistait à faire les TD d'un cours sur les modèles finis et les hiérarchie de complexité assuré par Jacques Mazoyer. J'ai beaucoup appris sur les jeux d'Ehrenfeucht Fraïssé et j'ai apprécié à certains moments l'indulgence des élèves qui ont eu, lors de l'évaluation finale, la gentillesse de reconnaître qu'ils avaient malgré tout beaucoup appris avec moi

Initiation à MAPLE aux agrégatifs, (1997-1998) le nouveau programme de l'agrégation de mathématiques comporte du calcul formel. J'ai donc monté un enseignement dans cette matière à base de travaux dirigés et adapté à ce public.

Diffusion des connaissances

J'ai participé chaque année à la Fête de la Science à Lyon :

- en 2001 : *Les protocoles de sécurité sur Internet et la logique de la connaissance*,
- en 2002 : *Science de l'information : science du matériel ou science de l'immatériel ?*
- en 2003 : *La logique de la connaissance et la connaissance sur le Web*,

– en 2004 : *Comment calculait-on il y a quatre mille ans chez les Babyloniens ?*

Le groupe séminaire est une activité particulièrement intéressante créée à l'initiative des élèves de l'ENS de Lyon. Il propose une à deux fois par mois une conférence de caractère interdisciplinaire pour une audience et un programme couvrant toutes les disciplines. J'y suis intervenu en novembre 2004 sur le thème *Je ne sais que tu ne sais pas, la logique de la connaissance*.

J'ai été interviewé en 2003 par la chaîne de télévision lyonnaise TLM sur la sécurité des cartes bancaires et du commerce électronique.

J'assure aussi des conférences aux étudiants : «au séminaire des élèves du MIM» (17 décembre 2002 : *Un calcul pour interpréter les séquents classiques*) et au cours de la semaine sport-études de la licence.

J'ai écrit trois articles dans la revue *La Recherche* : *Ces problèmes sans solution* en 2002, *Est-ce que « $P = NP$ » ?* en 2005 et « *Une idée reçue : les propositions vraies sont démontrables* » en 2007. Je participe activement à Wikipédia depuis novembre 2005, notamment à la refonte des articles sur la logique mathématique et l'informatique théorique.

ACTIVITÉS PROFESSIONNELLES SPÉCIFIQUES

Responsabilités scientifiques

Voici des responsabilités scientifiques que j'ai assumées ou que j'assume :

Membre du comité de programme des conférences internationales

RTA 85, RTA 87 (président), STACS 88, ESOP 89,
ALP 89, ESOP 90, LICS 93, RTA 96,
TLCA'97, PPDP 99, FOSSACS'2001, RTA-2004,
RTA-2005, TERMGRAPH'2006, ICALP'2007, AB'2008
ainsi que des workshops sur la terminaison WST'93, WST'95, WST'97,
WST'99, WST'01.

Membre du comité des conférences RTA (*Rewriting Techniques and Applications*), PPDP (*Principles and Practices of Declarative Programming*, jusqu'en 2003)

Membre du comité éditorial du journal *Applicable Algebra in Engineering, Communication and Computation*, publié par Springer Verlag depuis 1990.

Membre du groupe IFIP 1.6 *Term Rewriting* et **membre émérite du groupe IFIP 1.3** *Foundations of System Specification*.

Modérateur de la liste `rewriting@ens-lyon.fr` que j'ai créée en 1987 (252 abonnés).

Membre du groupe de nomination du Kyoto Prize depuis 1988.

Organisateur des conférences IFIP *Functional Programming and Computer Architecture* qui a eu lieu à Nancy en 1985, des journées du GDR Programmation en 1990, de l'École des jeunes chercheurs en Programmation en 1995 à Nancy et en 2000 à Lyon, du workshop international sur la Terminaison en 1995.

Membre du comité des thèses de l'ENS de Lyon (2001-2005)

J'ai de plus été quatre fois expert auprès de la National Science Foundation pour l'examen de contrats de recherche.

Membre d'associations

Je suis membre

- de l'Association for Computing Machinery (ACM),
- de l'European Association for Theoretical Computer Science (EATCS),
- de l'European Association for Programming Languages and Systems (EAPLS),
- de l'Association for Automated Reasoning (AAR)
- de l'Association Française d'Informatique Fondamentale (AFIF),
- de la Société des Personnels Enseignants et Chercheurs en informatique de France de (SPECIF).

Lecteur pour des revues

J'ai été sollicité comme *lecteur pour les revues suivantes* :

Acta Informatica,	Fundamenta Informaticae,
Information and Computation,	Journal of the ACM,
Journal of Automated Reasoning,	Journal of Computer and System Science,
Journal of Logic Programming,	Journal of Logic and Computation
Journal of Symbolic Computation,	Science of Computer Programming,
Techniques et Sciences de l'Informatique,	Theoretical Computer Science,
Transaction on Programming Languages and Systems	

ainsi que bien sûr que *Applicable Algebra in Engineering, Communication and Computation*, journal dont je suis éditeur, et de très nombreuses conférences internationales.

Rapporteurs de tenures, d'habilitations et de thèses

J'ai été rapporteur de

- 5 tenures aux États-Unis et en Allemagne,
- 10 habilitations à diriger des recherches en France,
- 35 thèses (au moins), dont 1 en Allemagne (Heidelberg), 1 en Belgique (Louvain), 1 en Norvège (Oslo), 2 aux Pays-Bas (Eindhoven et Utrecht), 1 en Pologne (Wrocław).

Responsabilités administratives

Voici des responsabilités diverses que j'ai assumées ou que j'assume.

D'évaluation

- Membre nommé du Conseil National des Universités (CNU), (1993-2000) (temps passé : 1 mois 1/2 an).
- Membre de la section 08 du Comité National du CNRS de 1983 à 1987 et à ce titre membre de comités de direction de laboratoires du CNRS (temps passé : 2 mois/an).

- Membre du comité de Département des Sciences Physiques pour l'Ingénieur (temps passé : 1 semaine/an).
- Président du comité des bourses de l'INRIA-Lorraine (1990-1994) (temps passé : 1 semaine/ an).
- Membre des jurys de recrutement des CR INRIA (Lorraine 1991-96, Sophia-Antipolis 1999 et 2004) et DR INRIA (1995, 2003), (temps passé : 1 semaine par jury).
- Membre du jury du concours d'entrée aux ENS, (1998, 2000, 2002) (temps passé : 1 mois par jury).
- Expert auprès du Comité National d'Évaluation (j'ai participé à l'évaluation du site d'Aix-Marseille [D-11]), (temps passé : 1 semaine).
- Conseiller auprès du président de l'Université de Perpignan pour une évaluation de l'informatique dans cette Université (2003), (temps passé : 1 semaine).
- Membre de commissions de spécialistes : Nancy (1983-1997), Lyon (1998-2003) (vice-président), Strasbourg (1992-1997), IJF Grenoble (1998-2001), INP Grenoble (2004,...).

Pédagogiques

- Directeur du DEA d'Informatique Fondamentale de Lyon (2001-2004) (temps passé : 1 semaine /an)
- Directeur adjoint de l'école doctorale de mathématique et informatique fondamentale de Lyon, (2001-2004). (temps passé : 1/2 semaine /an)
- Directeur adjoint, puis directeur, du département de mathématiques et d'informatique de l'ENS de Lyon, (1997-2001) (temps passé : 1 mois 1/2 /an)
- Membre du comité scientifique de de l'école doctorale *Informatique et Information pour la Société* de Lyon.
- Professeur à l'Université de Nancy 2 en 1983-1984 en remplacement de Jean-Pierre Jouannaud en séjour d'un an au SRI International. (temps passé : temps plein d'enseignement)

De recherche

- Responsable de l'équipe de recherche commune EURÉCA au CRIN⁶ et à l'INRIA-Lorraine, (1982-1997) (1 mois/an), 4 puis 8 permanents (temps passé : 1 mois 1/2 /an)
- Responsable de l'équipe de recherche PLUME du LIP (1997-...), 4 puis 5 permanents, (temps passé : 1 mois/an),
- Directeur du GDR *Programmation* (1992-1996) (temps passé : 1 mois/an).
- Directeur fondateur du Centre Charles Hermite, *Centre Lorrain de Compétence en Modélisation et Calcul à Haute Performance* (1993-1998) (temps passé : 1 mois 2/3 an).
- Membre de l'équipe de direction du Greco *Programmation*, (1983-1992) (temps passé : 5 jours/ an)

⁶EURÉCA a reçu la médaille d'argent du CNRS en 1987.

De management et de conseil

- Directeur adjoint du Centre de Recherche en Informatique de Nancy (UA CNRS 262), (1986-1992) (12 équipes, 200 personnes). (temps passé : 3 semaines /an)
- Membre du conseil d’administration de l’ENS de Lyon (1999-2003)
- Membre du conseil d’administration de SPECIF (2003-...)
- Membre du comité de pilotage du programme sécurité (SETIN) du l’Agence nationale de la recherche (2006-....).
- Membre du comité de pilotage du réseau Mathématiques de l’Informatique du CNRS (2002-2006).
- Membre du conseil de l’Université Nancy 2 (1973-1974).
- Président de la commission *Recherche* de SPECIF, (1989-1992)⁷. (temps passé : 1 semaine /an)
- Représentant de l’axe informatique au conseil scientifique de l’Institut Polytechnique de Lorraine, (1988-1992).
- Membre élu au comité consultatif régional de la recherche scientifique de la délégation Nord-Est du CNRS (1991-1994),
- Responsable de la mise en place du VAX et des moyens de calculs au CRIN (1983-1984). À cette occasion, j’ai réuni en 1984 les ingénieurs systèmes des différents sites pour créer la première interconnexion française entre ordinateurs de recherche en informatique⁸.

De gestion de projets

- Projet μ comm du Ministère de l’Industrie dans le cadre d’un projet OP-PIDUM avec la société NtSys de Lyon (1997-2001) (1MF, temps passé : un mois par an).
- Programme TELECOM du réseau RNTT (1997-2000), 180 kF (temps passé : 12 jours/an)
- BQR Installation ENS de Lyon (1997-1999), 200 kF (temps passé : dans le cadre de l’administration de l’équipe Plume).
- Projets *Lambda* (2000-2002), *Mikolaj* (2002-2004) et *Nicolas* (2004-2006), entre les universités de Cracovie et de Chambéry et l’ENS de Lyon, dans le cadre des projets Mira de la région Rhone-Alpes, organisation de visites et d’échanges, de colloques, suivis d’éditions d’actes [E-9] [E-10], (budget cumulé 80 k€, temps passé : quinze jours par an).
- Projet *Dullin* (2002-2003) soutenu par le programme CNRS *MathSTIC*, entre le laboratoire de mathématique de Chambéry et le LIP de l’ENS de Lyon (temps passé : 1 semaine).
- Responsable pour l’ENS de Lyon du cluster régional Rhône-Alpes EMSOC (2006-...) (temps passé : mineur).

⁷J’ai en particulier édité le compte-rendu des journées recherche de SPECIF [D-3] et l’annuaire des formations doctorales en informatique [E-6]

⁸Pour ceux qui n’ont pas connu cette période, cela peut paraître futile, mais en fait il s’agissait à cette époque héroïque d’un vrai travail d’innovation. J’ose affirmer qu’en créant cette interconnexion, j’ai contribué à mettre en place le précurseur d’Internet ente les laboratoires d’Informatique français.

Associatif

- Vice-président de SPECIF (1992-1994) (temps passé : 1 semaine /an).
- Président de SPECIF, (Société des Personnels Enseignants et Chercheurs de France) (2004-....) (temps passé : 1 mois /an).
- Membre du bureau d'INFORMATICS-EUROPE (2006-....), (temps passé : mineur).

LISTE DES PUBLICATIONS

Thèses

- T3** *Étude de quelques théories des langages et généralisation du théorème de Kleene*, Thèse de spécialité, Université de Nancy 1, (juin 71).
- TE** *Étude algébrique et relationnelle des types abstraits et de leurs représentations* Thèse d'État, Institut National Polytechnique de Lorraine, (septembre 79).

Livre (auteur)

- L-1** *Théorie des programmes*, Dunod, Paris, (1978). En collaboration avec Jean-Pierre Finance, Monique Grandbastien, Pierre Marchand, Roger Mohr, Alain Quéré, Jean-Luc Rémy. Accessible sur http://www.ens-lyon.fr/LIP/REWRITING/OLD_PUBLICATIONS_ON_LESCANNE/Livercy/.

Livre (traducteur)

- TR-1** *La maîtrise du développement du logiciel*, Édition d'organisation, Paris, (1990).
D'après le livre de Barbara Liskov et John Guttag, *Abstraction and Specification in Program Development*, En collaboration avec Jacques Guyard, Jean-Pierre Finance et Jacques Jaray.

Livres (éditeur)

- E-1** *Rewriting Techniques and Applications*, Lectures Notes in Computer Science 256, Springer Verlag (1987).
- E-2** *Rewriting Techniques and Applications*, Numéro spécial du journal Theoretical Computer Science, volume 67 (2 & 3), (1989).
- E-3** *Algebraic System Specification and Development : A Survey and Annotated Bibliography*, Lectures Notes in Computer Science volume 501, En collaboration avec M. Bidoit, H.J. Kreowsky, F. Orejas et D. Sanella.
- E-4** *Algebraic and Logic Programming* Akademie Verlag and Lectures Notes in Computer Science 343, Springer Verlag, (1989).

- E-5** *Algebraic and Logic Programming*, Numéro spécial du Journal of Logic Programming, volume 12, numéro 3, (1991).
- E-6** *Formations doctorales en Informatique*, CÉPADUÈS-Éditions, (1993).
- E-7** Valentin M. Antimirov (1961-1995). Theor. Comput. Sci. 155(2) : 289-290 (1996) présentation et édition d'un article post mortem de notre collègue, en collaboration avec Gregory Kucherov et Peter D. Mosses
- E-8** *Termination* Numéro spécial du Journal Applicable Algebra in Engineering, Communication and Computing, volume 12, numéros 1 et 2, (2001). En collaboration avec Jürgen Giesl
- E-9** *Workshop on Computational Logic and Applications 2004* Electronic Notes in Theoretical Computer Science, numéro 140.
- E-10** *Computational Logic and Applications, CLA '05* Discrete Mathematics & Theoretical Computer Science, in Vol. 2006.

Articles publiés dans des revues internationales (23)

- AR-1** *Equivalence entre la famille des ensembles réguliers et la famille des ensembles algébriques*, RAIRO Informatique Théorique / Theoretical Informatics, 10, 8 (1976), pp 57-81.
- AR-2** *On Multiset Orderings* Inf. Proc. Ltrs, 10, 2 (1982), pp 57-63, En collaboration avec Jean-Pierre Jouannaud.
- AR-3** *Modèles non déterministes de types abstraits*, RAIRO Informatique Théorique / Theoretical Informatics, 16, 3 (1982), pp 225-244.
- AR-4** *Some properties of the Decomposition Ordering, a simplification ordering to prove termination of rewriting systems*, RAIRO Informatique Théorique / Theoretical Informatics, 16, 4 (1982), pp 331-347.
- AR-5** *Behavioural Categoricity of Abstract Data Specifications*, The Computer Journal, 26, 4 (1983), pp 289-292.
- AR-6** ⁹ *Termination of Rewriting Systems by Polynomial Interpretations and its Implementation*, Science of Computer Programming, 9, 2, (1987), pp 137-160 En collaboration avec Ahlem Ben Cherifa.
- AR-7** *Tools for Proving Inductive Equalities, Relative Completeness and ω -Completeness* Information and Computation, 84, (1990), pp. 47-70. En collaboration avec Azzedine Lazrek et Jean-Jacques Thiel.
- AR-8** ¹⁰ *Equational problems and disunification* Journal of Symbolic Computation (1989), 3 & 4, pp. 371-426. En collaboration avec Hubert Comon.
- AR-9** *On the Recursive Decomposition Ordering with Lexicographical Status and other related Orderings* Journal of Automated Reasoning (1990), 6, pp. 39-49.

⁹D'après Citeseer, 3ème article le plus cité.

¹⁰2ème article le plus cité.

- AR-10** *Decidability of the Confluence of Finite Ground Term Rewriting Systems and of other related Term Rewriting Systems*. Information and Computation (1990), 88, pp. 187-201. En collaboration avec M. Dauchet, T. Heullard et S. Tison.
- AR-11** *Termination by Completion*, dans *Applicable Algebra in Engineering, Communication and Computation*, (1990), 1, pp. 79-96. En collaboration avec Françoise Bellegarde.
- AR-12** *The Term Rewriting Approach to Automated Theorem Proving*, Journal of Logic Programming (1992), 14, pp. 71-100. En collaboration avec J. Hsiang, H. Kirchner, P. Lescanne et M. Rusinowitch
- AR-13** *Well rewrite orderings and well quasi-orderings*, Journal of Symbolic Computation (1992), 14, pp. 419-435.
- AR-14** *On termination of one rule rewrite systems*, Theoretical Computer Science, (1994), 132, pp. 395-401.
- AR-15** *Elementary Interpretations in Proofs of Termination*, Formal Aspect of Computing, (1995), 7, pp. 77-90.
- AR-16**¹¹ *$\lambda\nu$, a calculus of explicit substitutions which preserves strong normalization*. J. of Functional Programming, Vol. 6, N 5 (septembre 1996). En collaboration avec Zine-el-Abidine Benaïssa, Daniel Briaud et Jocelyne Rouyer.
- AR-17** *On strong normalization of explicit substitution calculi* J. of Functional and Logic Programming, 2000 (7). En collaboration avec Frédéric Lang
- AR-18** *Reductions, intersection types, and explicit substitutions* Mathematical Structures in Computer Science 13 (1) : 55-85 (2003). En collaboration avec Daniel Dougherty. Version étendue de [CI-33]
- AR-19** *Intersection types for explicit substitutions*, Information and Computation, Vol 189 (1) pp 17-42, (2004). En collaboration avec Stéphane Lengrand, Dan Dougherty, Mariangiola Dezani-Ciancaglini et Steffen van Bakel
- AR-20** *Addressed Term Rewriting Systems : Application to a Typed Object Calculus* Mathematical Structures in Computer Science, Volume 16, Issue 04, August 2006, pp 667-709, En collaboration avec Dan Dougherty et Luigi Liquori
- AR-21** *Mechanizing common knowledge logic using COQ*, Annals of Mathematics and Artificial Intelligence, Volume 48, Numbers 1-2, September, 2006 pp. 15-43
- AR-22** *Computation with Classical Sequents*. Mathematical Structures in Computer Science 18(3) : 555-609 (2008). En collaboration avec S. van Bakel
- AR-23** *Characterizing strong normalization in the Curien-Herbelin symmetric lambda calculus : extending the Coppo-Dezani heritage*, Theor. Comput. Sci. 398(1-3) : 114-128 (2008) (special issue Festschrift Coppo, Dezani, Ronchi). (En collaboration avec D. Dougherty et S. Ghilezan)

¹¹Article le plus cité.

Communications à des conférences internationales (35)

- CI-1** *Quelques applications des classes équationnelles conformes* 2nd colloque sur les arbres en algèbre et programmation (CAAP), (Arnold Ed.), (1977), pp 198-212.
- CI-2** *Manipulation d'arbres syntaxiques et construction de programmes*, 5ème colloque sur les arbres en algèbre et en programmation (CAAP), (Dauchet Ed.), (1980). En collaboration avec Jacques Guyard.
- CI-3** *Decomposition Ordering as a Tool to prove the termination of Rewriting Systems* 7th IJCAI, Vancouver Canada, (1981), pp 548-550.
- CI-4** *Two implementations of Recursive Path Ordering on Monadic Terms*, 19th Annual Allerton Conf. on Communication, Control and Computing, Allerton House, Monticello, Illinois, (1981), pp 634-643.
- CI-5** *Recursive Decomposition Ordering*, IFIP Conf. on Formal Description of Programming Concepts II, Garmish Partenkirchen (1981), (D. Bjørner Ed.) North-Holland, pp 331-346. En collaboration avec Jean-Pierre Jouannaud et F. Reinig.
- CI-6**¹² *Computer Experiments with the REVE Term Rewriting System Generator* 10th ACM Symposium on Principles of Programming Languages (POPL), January 24-26 1983, Austin Texas, pp 99-108.
- CI-7** *On the study of Data Structures : Binary Tournaments with Repeated Keys* 10th International Conference on Automata Languages and Programming (ICALP), Lecture Notes in Computer Science 154 (1983), pp 466-477. En collaboration avec Jean-Marc Steyaert.
- CI-8** *Uniform Termination of Term Rewriting Systems : Recursive Decomposition Ordering with Status* Conférence sur les Arbres en Algèbre et en Programmation (CAAP), Bordeaux (mars 1984), Cambridge University Press, Cambridge, pp. 181-194.
- CI-9** *Term Rewriting Systems and Algebra* Conference on Automated Deduction (CADE), Lectures Notes in Computer Sciences 170. (Napa Valley, mai 1984).
- CI-10** *NARROWER : a new procedure for solving equations and its application to logic programming* Actes de la première conférence internationale Rewriting Techniques and Applications (RTA), Lectures Notes in Computer Science 202, Springer Verlag. (Dijon, mai 1985). En collaboration avec Pierre Réty, Claude Kirchner et Hélène Kirchner.
- CI-11***¹³ *An Introduction to Rewriting* Support de cours de l'école d'été internationale Advanced Programming Technologies, (San Sebastien, Espagne, septembre 1985)

¹²Deuxième conférence la plus citée.

¹³Les références marquées d'une étoile correspondent à des conférences invitées.

- CI-12** *An actual implementation of a procedure that mechanically proves termination of rewriting systems based on inequalities between polynomial interpretations*, Proc. 8th Conf. on Automated Deduction (CADE), Oxford (GB), Lecture Notes in Computer Science, 230, Springer Verlag (1986), pp 42-51. En collaboration avec Alhem Ben Cherifa.
- CI-13** *Proving Termination of Associative Rewriting Systems by Rewriting*, Proc. 8th Conf. on Automated Deduction (CADE), Oxford (GB), Lecture Notes in Computer Science, 230, Springer Verlag (1986), pp 52-61. En collaboration avec Isabelle Gnaedig.
- CI-14** *Transformation Ordering*, CAAP '87, (Pise, Italie, Mars, 1987), Lecture Notes in Computer Science, Springer Verlag (1987), En collaboration avec Françoise Bellegarde
- CI-15** *Decidability of the Confluence of Ground Term Rewriting Systems* 2nd IEEE Symposium on Logic in Computer Science (LICS), (Ithaca, New-York, Juin 1987). En collaboration avec Max Dauchet, Thierry Heuillard et Sophie Tison.
- CI-16** *Solving Disequations* 2nd IEEE Symposium on Logic in Computer Science (LICS), (Ithaca, New-York, Juin 1987). En collaboration avec Claude Kirchner
- CI-17*** *Current Trends in Rewriting Techniques and Related Problems*, IBM International Symposium on Current Trends in Computer Algebra, Bad-Neuahr, République Fédérale d'Allemagne, Lecture Notes in Computer Science 296, Springer Verlag (1986), pp 38-51.
- CI-18*** *Completion Procedures as Transition Rules + Control*, conférence invitée à Colloquium Current Concept In Programming Languages, TAP-SOFT, Barcelone (Espagne) Lecture Notes in Computer Science 351, Springer Verlag (1989), pp 28-41.
- CI-19** *Well Rewrite Orderings*, 5th IEEE Symposium on Logic in Computer Science (LICS), (Philadelphie, Pennsylvanie, Juin 1990), pp 249-256.
- CI-20*** *Implementation of Completion by Transition Rules + Control : ORME*, conférence invitée à la seconde conférence internationale Algebraic and Logic Programming (ALP), Nancy, Lecture Notes in Computer Science 463, Springer Verlag (1990), pp 262-269.
- CI-21*** *Rewrite orderings and termination of rewrite systems*, conférence invitée à au seizième Symposium International Mathematical Foundations of Computer Science (MFCS), Kazimierz Dolny (Pologne), Lecture Notes in Computer Science 520, Springer Verlag (1991), pp 17-27.
- CI-22** *Polynomial Interpretations and the Complexity of Algorithms* Proc. 11th Conf. on Automated Deduction (CADE), Saratoga Springs (Etats-Unis), Lecture Notes in Computer Science 607, Springer Verlag (1992), pp 139-147. En collaboration avec Adam Cichon.
- CI-23** *Termination of rewrite systems by elementary interpretations* Proc. Third International Conference on Algebraic and Logic Programming (ALP), Pise (Italie), Lecture Notes in Computer Science 632, Springer Verlag (1992), pp 21-36.

- CI-24** ¹⁴ *From $\lambda\sigma$ to $\lambda\nu$, a journey through calculi of explicit substitutions* 21st ACM Symposium on Principles of Programming Languages (POPL), 16-19 Janvier 1994, Portland, Oregon, pp 60-69.
- CI-25** *Explicit Substitutions with de Bruijn's Levels*. Actes de la cinquième, conférence internationale Rewriting Techniques and Applications (RTA), Lecture Notes in Computer Science 914, Springer Verlag (1995), pp 294-308. En collaboration avec Jocelyne Rouyer.
- CI-26** *A Formal proof of a protocol for communication over faulty channels using the Larch Prover*. Proc. of 8th Int. Conf. on Formal Descriptions Techniques for Distributed Systems and Communications Protocols, FORTE'95 (16 pages). Comme collaborateur de Boutheina Chetali.
- CI-27** *Modeling Sharing and Recursion for Weak Reduction Strategies using Explicit Substitution*. Eighth International Symposium on Programming Languages, Implementations, Logics, and Programs (PLILP). Vol. 1140 in LNCS, pages 393-407, Aachen, Germany, September 1996. Springer-Verlag. En collaboration avec Zino Benaïssa et Kristoffer Rose.
- CI-28** *A Framework for Defining Object-Calculi* World Congress on Formal Methods (FM'99) 20-24 September 1999, Toulouse, France , Springer LNCS vol. 1709, pp. 963-982,. En collaboration avec Frédéric Lang et Luigi Liquori.
- CI-29** *Reductions, intersection types, and explicit substitutions* Typed lambda calculi and applications (TLCA), mai, 2001, Cracovie, Pologne, Springer, Lecture Notes in Computer Science, 2044, 2001, pp 121-135 En collaboration avec Daniel Dougherty.
- CI-30** *An improved system of intersection types for explicit substitutions*, Conference on Foundations of Information Technology in the era of Network and Mobile Computing, IFIP Congress, Montréal, August 2002, pp. 511-524, Kluwer Academic Publishers, En collaboration avec Daniel Dougherty et Stéphane Lengrand.
- CI-31** *Classical proofs and typed processes : intersection types and strong normalization* TYPES 03 workshop, publié après sélection dans Springer LNCS vol. 3085, pp 226-241 En collaboration avec Silvia Ghilezan.
- CI-32** *Characterizing strong normalization in a language with control operators*, PPDP 2004, 6th ACM-SIGPLAN International Conference on Principles and Practice of Declarative Programming, pp 155-166 En collaboration avec Silvia Ghilezan et Daniel Dougherty.
- CI-33** *The language X : circuits, computations and Classical Logic*, ICTCS 2005 Ninth Italian Conference on Theoretical Computer Science, Certosa di Pontignano (Sienne), Italie, 12-14 Octobre 2005 En collaboration avec Steffen van Bakel et Stéphane Lengrand
- CI-34** *Strong normalization of the dual classical sequent calculus*, Logic for Programming, Artificial Intelligence, and Reasoning, 12th International

¹⁴Conférence la plus citée.

Conference, LPAR-2005, Lecture Notes in Computer Science 3835 Springer 2005, 169-183 En collaboration avec Daniel Dougherty, Silvia Ghilezan et Silvia Likavec

CI-35 *Rewriting Game Theory as a Foundation for State-Based Models of Gene Regulation* Computational Methods in Systems Biology, Lecture Notes in Computer Science Publisher Springer Berlin / Heidelberg ISSN 0302-9743, Vol. 4210/2006, pp. 257-270 En collaboration avec C. Chettaoui, F. Delaplace M. Vestergaard, R. Vestergaard

Réunions de travail internationales avec publication des actes (12)

WK-1 *An exercise in LP, the proof of a non restoring division circuit*, Proc. of First International Workshop on Larch, U. Martin et J. Wing ed., Springer Verlag, Workshop in Computing. Comme collaborateur de Boutheina Chetali.

WK-2 *Verification and Programming of First-order Unification, in the Calculus of Constructions with Inductive Types* UNIF'93, Résumé étendu (6 p.). Comme collaborateur de Joseph Rouyer.

WK-3 *Improving Abstract Data Type Specifications by an Appropriate Choice of Constructors*, Proc. International Workshop on Program Construction. (Bonas, France, septembre 1980). En collaboration avec Christine Choppy et Jean-Luc Remy.

WK-4* *The lambda calculus as an abstract data type*, conférence invitée à Recent Trends in Data Type Specification, 11th Workshop on Specification of Abstract Data Types Joint with the 8th COMPASS Workshop, Oslo, Norway, September 19-23, 1995. Publié dans Springer, Lecture Notes in Computer Science, vol. 1130, pp 74-80.

WK-5 *Formally validated specification of a micro-payment protocol*, présenté au workshop SAVE 2001 Specification, Analysis and Validation for Emerging Technologies in Computational Logic, associé à ICLP 01 En collaboration avec P. Dargenton, D.Hirschhoff et E. Pommateau.

WK-6 *Explicit substitutions and intersection types*, WST'03, Workshop on Termination, (2003)

WK-7 *Pure type systems, cut and explicit substitutions*, HOR'04, 2nd International Workshop on Higher-Order Rewriting, Wednesday June 2, 2004 Aachen, Germany. En collaboration avec Romain Kervarc.

WK-8 *Solving Equations in a Language With Control Operators* 18th International Workshop on Unification Cork, Ireland, July 5, 2004 En collaboration avec Stephane Le Roux.

WK-9 *Addressed Term Rewriting Systems* TERMGRAPH 2004, 2nd International Workshop on Term Graph Rewriting Rome, Italy, 2 October, 2004. En collaboration avec Daniel Dougherty, Frederic Lang et Luigi Liquori

WK-10 *Experiments in higher order epistemic logic with common knowledge using a proof assistant*. WPLHG05 : Workshop on Programming Logics in memory of Harald Ganzinger, Saarbrücken, June 3 and 4, 2005.

- WK-11** *Intersection and Union Types in the lambda-my-my/tilde-calculus*. Proceedings of the Third International Workshop on Intersection Types and Related Systems (ITRS 2004), Electr. Notes Theor. Comput. Sci. 136 : 153-172 (2005) En collaboration avec Daniel Dougherty et Silvia Ghilezan
- WK-12** *Erasure and Duplication in Classical Computation* Journées françaises sur les langages applicatifs, JFLA 2007, 27-30 janvier 2007 Aix-les-Bains. En collaboration avec Dragiša Žunić

Articles publiés dans des revues nationales (3)

- AN-1** *La Réécriture : Motivations, Théorie, Mise en Oeuvre et Applications, un présentation du logiciel REVE*, Enjeux 61, Septembre 1985, pp. 71-75. En collaboration avec Jean-Pierre Jouannaud.
- AN-2** *La Réécriture*, Techniques et Sciences Informatiques, AFCET-Gauthier-Villars, 5, 6, (1986), pp. 433-452. En collaboration avec Jean-Pierre Jouannaud.
- AN-3** *Explicit substitutions and intersection types*. Schedae Informaticae vol.12, 2003.

Communications à des conférences nationales (10)

- CN-1** *Trois approches sémantiques d'un langage permettant l'écriture des programmes analysés déductivement*, Congrès AFCET Informatique (1978). En collaboration avec Jean-Pierre Finance.
- CN-2** *MEDEE : a type of language for the deductive programming method*, Workshop on Reliable Software (Bonn 78) (Raulefs Ed.), Carl Hauser Verlag, Munchen-Wien, (1979), pp 63-72. En collaboration avec le groupe Castor.
- CN-3** *L'utilisation d'un système d'édition et de manipulation de programmes comme support d'une méthode de programmation*, Congrès AFCET Regards sur l'Informatique, (1980). En collaboration avec Jacques Guyard.
- CN-4** *A constructive proof of the well-foundedness of the Recursive Path Ordering*, Colloque AFCET les Mathématiques de l'Informatique, (1982).
- CN-5** *Une introduction au langage CLU*, Journées Langages Orientés Objets, (Cap d'Agde, 1984)
- CN-6** *Les outils de preuves de terminaison dans REVE*, Journées sur la logique et l'informatique, 29-30 avril 1986, B. Courcelle Ed., rapport UER de mathématique et informatique de l'Université de Bordeaux 1, vol. I-8620, pp 20-34.
- CN-7** *Programmation fonctionnelle et substitutions explicites*. CARI'96, 3^{eme} Colloque Africain sur la Recherche en Informatique, Libreville 9-16 octobre 1996.

- CN-8** *Algèbres de mots et algèbre universelle*, Colloque AFCET-SMF, (1978), pp. 147-156.
- CN-9*** *Calculi of explicit substitutions : new results*, conférence invitée à la conférence APPIA-GULP-PRODE'96. En collaboration avec Zine-el-Abidine Benaïssa et Daniel Briaud.

Descriptions de Logiciels

- DL-1** *REVE a Rewrite Rule Laboratory*, Proc. 8th Conf. on Automated Deduction, Oxford (England), Lecture Notes in Computer Science, 230, Springer Verlag (1986), pp 696-697.
- DL-2** *ORME, an Implementation of Completion Procedures as Sets of Transitions Rules* Proc. 10th Conf. on Automated Deduction, Kaiserslautern (RFA), Lecture Notes in Artificial Intelligence, Springer Verlag (1990), pp 661-662.
- DL-3** *An introduction to ORME* Rapport CRIN 94-R-164.

Rapports Internes et Notes

- RI-1** *Le λ -calcul et la logique combinatoire*, Notes, (1973).
<http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/lambda.html>
- RI-2** *Aspect de la théorie de la Programmation*, Support de cours de l'École d'Été de Tarbes (1974).
- RI-3** *A Well-Founded Recursively Defined Ordering on First Order Terms* (1980), CRIN 80-R-005. En collaboration avec F. Reinig.
- RI-4** *Une introduction à Mentor* (1980), En collaboration avec Jacques Guyard.
- RI-5** *Analysis of Data Structures with non Distinct Keys* (1982), CRIN 82-R-039.
- RI-6** *Initiation à la complexité des algorithmes* (1985). En collaboration avec Jean-Luc Rémy, François Schwaab et les étudiants de DEA de la promotion 1982-1983.
- RI-7** *REVE : A Rewrite Rule Laboratory* texte de la communication invitée présentée au workshop on Formal Software Development (Nyborg, Danemark, mai 1984).
- RI-9** *Proving Inductive Equalities, Algorithms and Implementations*, une version préliminaire de [AR-7]. En collaboration avec Azzedine Lazrek et Jean-Jacques Thiel.
- R-10** *Verification and Programming of First-order Unification, in the Calculus of Constructions with Inductive Types* Journal of Logic and Computation, (1993), à paraître jamais
- RI-15** *Proving identities in commutative semigroups and monoids : an automaton approach*

- RI-17** *The Calculus of Explicit Substitutions* λv Mars 1994, Rapport INRIA 2-2222 et CRIN 94-R-043. En collaboration avec Jocelyne Rouyer.
- RI-19** *De Bruijn's $C\lambda\xi\phi$ calculus of explicit substitutions revisited.*
- RI-20** *Triad Machine : A General Computational Model for the Description of Abstract Machines* Rapport CRIN 95-R-410.
- RI-22** *Epistemic Logic in Higher Order Logic An experiment with COQ.* Rapport LIP RR2001-12.
- RI-34** *A Generic Object-Calculus Based on Addressed Term Rewriting Systems* En collaboration avec Daniel Dougherty, Frédéric Lang, Luigi Liquori et Kristoffer Rose.
- RI-35** *Explicit Pure Type Systems for the Lambda-Cube.* Rapport LIP RR2004-08, February 2004 En collaboration avec Romain Kervarc.
- RI-36** *CP games : a tutorial,* Rapport de recherche LIP, RR2006-51, Dec, 2006

Divers et Articles Historiques

- D-1** *Divergence of the Knuth-Bendix procedure and termination orderings,* Bull. EATCS, 30, Octobre 1986, pp-80-83
- D-2** *Well quasi-orderings in a paper by Maurice Janet,* Bull. EATCS, 39, Octobre 1989, pp 185-188.
- D-3** *Journées recherche de SPECIF,* TSI, volume 10, n°3, (1991), pp. 225-236.
- D-4** *Rewriting Techniques and Applications – RTA91,* Comptes-rendus de la conférence publiés dans *Sigact News*, 22, pp-24-30 et le Bulletin de l'EATCS En collaboration avec Héléne Kirchner
- D-5** *La science informatique,* Encyclopédie Illustrée de la Lorraine, Histoire des Sciences et des Techniques, Volume *Sciences exactes*, Éditions Serpenoises, pp 105-116.
http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/histoire_lorraine.
- D-6** *Rapport d'activités du Centre Charles Hermite 1994.*
- D-7** *Rapport d'activités du Centre Charles Hermite 1995.*
- D-8** *Proposition au CNRS de constitution d'un GDR Programmation,* (1993).
- D-9** *Rapport d'activités du GDR-PRC Programmation.* (1996).
- D-10** *Rapport d'activités du Centre Charles Hermite 1996.*
- D-11** *Le site universitaire d'Aix-Marseille,* rapport d'évaluation du Comité National d'Évaluation (juin 2001) accessible par W^3 à l'adresse
http://www.cne-evaluation.fr/WCNE_pdf/Aix-Marseille_Site.pdf.
- D-12** *Automated Deduction as an international discipline – An open letter to the president of CADE* Association for Automated Reasoning Newsletter 35, (Dec 1996).
- D-39** *Ces problèmes sans solution,* La Recherche, N°351 - mars 2002, Rubrique B.A.-BA, pp. 54-55, En collaboration avec Christophe Genolini.

- D-40** *Est-ce que tous les problèmes ont une résolution par programme ? Une illustration par les holorimes, voir*
<http://perso.ens-lyon.fr/pierre.lescanne/TEXTS/baba351.html>
complément de l'article précédent.
- D-42** *Est-ce que « $P = NP$ » ?* Les dossiers de La Recherche, N°20, Août-Octobre 2005. En collaboration avec Nicolas Hermann
- D-44** *Alfred Tarski : Life and Logic by Anita Burdman Feferman and Solomon Feferman* revue du livre pour ACM SIGACT News Vol 37, No 1, 2006. La version française est parue dans le Bulletin de SPECIF N°54 de décembre 2005.
- D-45** *A Collection of Hardly Accessible Publications in Theoretical Computer Science with a Historical Interest*, document accessible par W^3 à l'adresse
<http://perso.ens-lyon.fr/pierre.lescanne/NotAccessible.html>
- D-46** *Les propositions vraies sont démontrables*, numéro spécial de la revue *La Recherche* sur les idées reçues en science.

Documents sur WWW

- Une liste tenue à jour se trouve à
<http://perso.ens-lyon.fr/pierre.lescanne/transparency.html>.
- WWW-1** *Description du Centre Charles Hermite* document accessible par W^3 à l'adresse
<http://http://cch.loria.fr/>
- WWW-2** *Description du GDR Programmation*
- WWW-3** *Comment candidater ?*, document à l'intention des doctorants, disponible sur
<http://perso.ens-lyon.fr/pierre.lescanne/TEXTS/candidater.html>.
- WWW-4** *On the relations between rewrite systems, abstract data types and lambda-calculus*, accessible par W^3 à l'adresse
<http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/oslo.pdf>.
- WWW-5** *Substitutions explicites et forte normalisation*, accessible par W^3 à l'adresse
http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/ami_lyon.ps.gz
- WWW-6** *Explicit substitutions*, accessible par W^3 à l'adresse
<http://perso.ens-lyon.fr/pierre.lescanne/PUBLICATIONS/GULP.ps.gz>
- WWW-7** *Enseignement de Pierre Lescanne*, accessible par W^3 à l'adresse
<http://perso.ens-lyon.fr/pierre.lescanne/enseignement.html>
- WWW-8** *Histoire des algorithmes*, accessible par W^3 à l'adresse
<http://perso.ens-lyon.fr/pierre.lescanne/ENSEIGNEMENT/histoire.html>

DESCRIPTION DÉTAILLÉE DES ACTIVITÉ DE RECHERCHE

Le domaine de recherche auquel j'ai consacré l'essentiel de mon activité scientifique est *l'informatique fondamentale*. J'ai désiré m'attaquer, en effet, aux problèmes qui sont au cœur de *l'informatique* et de *l'intelligence augmentée*; pour cela j'utilise et perfectionne des concepts *mathématiques* et tente de faire progresser la *logique*. Parallèlement, je mets en œuvre, sous forme de *logiciels*, des outils qui découlent de mes réflexions et qui doivent pouvoir servir à d'autres chercheurs. Jusqu'à présent, mes travaux ont porté essentiellement sur neuf axes qui forment une succession logique dans mon approche des principaux problèmes de cette discipline. Cette succession correspond à peu près à la chronologie. Ces neuf axes sont :

- les langages formels,
- les spécifications algébriques,
- la réécriture,
- le logiciel REVE,
- le logiciel ORME,
- la déduction automatique appliquée aux protocoles,
- l'application de la réécriture à la programmation fonctionnelle, à la programmation objet et aux assistants de preuve, à travers la théorie des substitutions explicites,
- les applications au commerce électronique,
- l'interprétation du calcul des séquents classiques.

Les Langages Formels (69-72)

Au moment où j'ai commencé mes recherches sous la direction de Claude Pair [D-5]¹⁵, la principale préoccupation des informaticiens était de construire des compilateurs et une grande partie de l'activité dans ce domaine tournait autour des langages formels. Dans ce courant, ma thèse de spécialité [T3] a porté sur l'étude d'un problème de cette théorie, à savoir l'extension du théorème de Kleene à des algèbres autres que les algèbres monadiques, cela a nécessité tout d'abord la généralisation des concepts d'ensembles algébriques, c'est-à-dire

¹⁵Les références se rapportent à la listes des publications

plus petites solutions d'une équation à point fixe dans l'ensemble des parties et d'ensembles réguliers ou rationnels, c'est-à-dire calculés par itération. Pour pouvoir généraliser ces concepts, il faut être capable d'étendre la structure d'algèbre à l'extension ensembliste naturelle de celle-ci. Cette propriété, dite de P -stabilité, n'est possible que sous certaines conditions sur les équations définissant la variété. Par le fait qu'elle étudiait des algèbres générales, cette étude préfigurait celles qui eurent lieu plus tard sur les schémas de programmes par l'intermédiaire des magmas (un autre nom des algèbres) et sur les spécifications algébriques de types abstraits, j'étais aussi parmi les premiers chercheurs à introduire les catégories en informatique théorique. J'ai, par la suite, étendu la P -stabilité aux algèbres hétérogènes, sous-tendant ainsi le calcul non-déterministe sur les types abstraits [AR-3]. Le théorème principal de ma thèse de spécialité affirme l'équivalence entre les ensembles algébriques et les ensembles réguliers. Ces travaux ont donné lieu à une publication [AR-1]. La communication intitulée *Algèbres de mots et algèbre universelle* [CN-8] illustre tous ces concepts sur l'ensemble des mots et constitue sur le sujet une synthèse commentée d'exemples.

Les Spécifications Algébriques (73-80)

Par la suite, mon intérêt s'est dirigé vers les problèmes de spécification et sur l'outil qui me paraissait, à l'époque, être le plus adapté pour cela, les *types abstraits de données*. Dans ma thèse d'État [TE], on trouve deux tentatives pour relier des modèles de calcul et les spécifications algébriques de types abstraits; l'un de ces modèles reprend et reformule une approche qui avait été faite à Nancy par Claude Pair et Jean-Luc Rémy sur les *structures de données* préfigurant longtemps avant l'heure ce que l'on appellerait aujourd'hui le *modèle objet*, l'autre utilise le cadre du calcul relationnel, aujourd'hui tout-à-fait d'actualité dans l'étude des programmes indéterministes. La deuxième approche a mis en évidence la complexité de certaines preuves qui peuvent devenir très fastidieuses. J'ai alors entrepris la construction d'un démonstrateur de théorème pour l'algèbre relationnelle. Par la suite, ce système est devenu un démonstrateur pour les théories équationnelles, en général. Il était inspiré de PROLOG, encore à ses débuts à l'époque, 1977. Écrit en Pascal (un langage révolutionnaire pour l'époque) il ressemblait pour ainsi dire à un Prolog manipulant des équations entre termes. J'ai ainsi obtenu certains résultats, présentés dans ma thèse, à propos des groupes commutatifs, par exemple. Très vite, j'ai entrevu les limites d'une telle approche, si elle n'est pas appuyée sur une théorie solide. Les systèmes de réécriture me sont alors apparus comme l'outil qu'il fallait et cela a décidé de ma recherche actuelle. Ma thèse comporte aussi une étude des spécifications algébriques de types abstraits pour elles-mêmes, j'y ai montré l'importance de l'observation des objets uniquement par leur comportement externe, mon intérêt s'est alors porté sur les algèbres finales, c'est-à-dire les algèbres où deux objets sont égaux si et seulement si ils ont même comportement. Le problème se pose alors de savoir si la spécification capture cette égalité de comportement. Si une spécification possède cette propriété, on dit qu'elle est

catégorique (au risque d'être un peu lourd et pour être plus précis, on peut dire qu'elle a la propriété de catégoricité comportementale). Cette étude a été reprise dans un article et j'ai montré qu'on ne peut pas décider de la catégoricité d'une spécification algébrique [AR-5]. Comme je l'ai dit plus haut, j'ai consacré un chapitre de ma thèse aux modèles non déterministes de spécifications de types abstraits de données, ceci a été repris dans [AR-3]. D'autre part, avec Christine Choppy et Jean-Luc Rémy nous avons esquissé une méthodologie de spécifications de types abstraits; elle est fondée sur la recherche de certaines opérations dites constructeurs qui permettent d'exprimer tous les objets [WK-3, CH-1], ces articles contiennent aussi les germes d'une étude de complexité reprise plus tard par Christine Choppy, Stéphane Kaplan et Michèle Soria.

La Réécriture (80-92)

Mes travaux sur les spécifications algébriques de types abstraits m'ont convaincu de la nécessité des preuves automatiques. En effet, malgré tout le soin que l'on peut leur apporter, celles-ci demandent une telle précision que l'aide d'un ordinateur s'impose. D'autre part, elles ont un aspect répétitif et s'intègrent dans un cadre bien défini puisqu'il s'agit de preuves dans des théories équationnelles. En fait, comme cela m'était apparu lors de ma première tentative, les choses ne sont pas simples, et l'étude des systèmes de réécriture de termes déjà esquissée dans ma thèse, devait être poursuivie.

De la complétion à la preuve de terminaison. Le principal outil sur lequel s'appuyer a été fourni par Knuth qui a donné un algorithme pour vérifier si un système de réécriture est aussi puissant, du point de vue de ce qu'il peut prouver, que le système d'équations qui lui a donné naissance. Cet algorithme teste une propriété dite de confluence locale, or la propriété importante est celle de *confluence* qui signifie que le résultat d'un calcul dans le système de réécriture est indépendant du chemin suivi. La confluence est conséquence de la confluence locale à condition que le système soit à terminaison uniforme ou noethérien. De plus, au prix d'une légère modification, l'algorithme de Knuth et Bendix peut transformer un système de réécriture non confluent, construit par exemple à partir d'un système d'équations, en un système confluent qui peut donc être utilisé pour prouver plus efficacement les identités qui sont conséquences des équations. Dans ce cas, l'algorithme (on devrait dire la procédure) de Knuth et Bendix engendre des équations qu'il faut orienter en règle de réécriture tout en conservant la terminaison uniforme du système. Pour toutes ces raisons, la terminaison uniforme des systèmes de réécriture m'est apparue comme un problème central. Or les solutions au moment où je commençais à m'y attaquer étaient inadéquates. Un rapport de David Plaisted de l'Université d'Illinois à Urbana-Champaign, semblait cependant ouvrir une voie prometteuse, il introduisait un nouvel ordre sur les termes fondé sur une *précédence*, nom sous lequel j'appelle un ordre sur les symboles. J'ai tenté d'en ré-exprimer les principales idées et j'ai invité Fernand Reinig à en faire son thème de recherche. La première étape m'a conduit à redonner une définition plus aisée à comprendre, retrouvant ainsi

indépendamment de Dershowitz l'ordre récursif sur les chemins, ensuite j'ai essayé de prouver la bonne fondation introduisant le concept de *décomposition* [RI-3]. Ce concept sera à l'origine d'un nouvel ordre [CN-4, AR-4] étendu par Fernand Reinig au cas d'une précedence partielle, Jean-Pierre Jouannaud et moi avons complété les preuves de Fernand Reinig et en collaboration avec lui en avons proposé la définition dans [CI-5]. Un intérêt de l'ordre de décomposition est une propriété appelée *incrémentation* qui permet d'augmenter la précedence au vol durant l'exécution de l'algorithme de Knuth et Bendix. Par conséquent, il n'est absolument pas nécessaire d'indiquer les critères utilisés dans la preuve de terminaison au moment du lancement, les choix peuvent être faits en cours de route en suivant les suggestions proposées. Cette possibilité a été implantée dans le système REVE dont elle constitue l'une des principales originalités [CI-6]. Dans [CI-3], j'ai présenté l'ordre de décomposition dans le cas d'une précedence totale. Ensuite, j'ai ajouté une autre possibilité que l'ordre de décomposition n'avait pas, la prise en compte des règles de réécriture apparentées à l'associativité [CI-8, AR-9]. Il devient ainsi un ordre plus général et plus puissant que l'ordre récursif sur les chemins de Dershowitz et que son extension proposée par Kamin et Lévy, classiquement cités pour ce genre d'utilisation. Durant l'été 1984, au cours d'un séjour de trois mois au MIT, j'ai considérablement amélioré la manière dont REVE fait les suggestions, tant en ce qui concerne la précedence qu'en ce qui concerne les statuts. Maintenant celles-ci sont si précises qu'une automatisation complète est possible.

La terminaison. Par la suite, j'ai continué à travailler sur les *ordres de preuve de terminaison*. Je suis parti de la constatation suggérée par Françoise Bellegarde au cours de ses recherches qu'il existe des systèmes de réécriture, surtout en transformation de programmes, qui résistent aux ordres de la famille des ordres de décomposition et j'ai défini, avec Ahlem Ben Cherifa, une méthode pour prouver la terminaison fondée sur des *interprétations polynomiales* [CI-12, AR-6]. Cette méthode a été mise en œuvre dans REVE et dans ORME et se trouve être très efficace et suffisamment générale, comme l'ont montré de nombreux exemples que nous avons traités. Et dans ce cadre, j'ai caractérisé les interprétations polynomiales des opérateurs associatifs et commutatifs. Les interprétations polynomiales ont constitué, pendant dix ans, la seule méthode implantée qui fonctionne dans ce cas. Nous avons cependant Adam Cichon et moi [CI-22] montré leur limitation en prouvant que si la définition d'une fonction par un système de réécriture est prouvée terminer grâce à une interprétation polynomiale, alors cette fonction a une croissance au plus polynomiale. Cela m'a donné l'idée d'une extension de la méthode des interprétations polynomiales à des *interprétations élémentaires* c'est-à-dire à des interprétations construites en utilisant les opérations des polynômes, à savoir la somme et le produit, auxquelles on ajoute l'exponentielle, j'ai ainsi proposé une implantation que j'ai en suite mis en œuvre en CAML [CI-23, AR-15]. J'ai travaillé avec Isabelle Gnaedig à une extension de l'ordre récursif sur les chemins au cas associatif et commutatif [CI-13]. Avec Françoise Bellegarde, j'ai étudié un ordre pour la terminaison appelé *ordre de transformation* [CI-14, AR-11] où les transformations

sont décrites par des systèmes de réécriture. Une des faiblesses des ordres de simplification qui sont les plus utilisés tient au fait qu'ils étendent le plongement de Kruskal et Higman, un ordre comparant les termes par effacement de symboles ; de ce fait, ils ne peuvent pas prouver la terminaison de certains systèmes. J'ai proposé une *extension du plongement* et donc une extension du théorème de Higman pour essayer de circonvenir cet inconvénient [CI-19, AR-13]. Max Dauchet a proposé une preuve de l'indécidabilité de la terminaison des systèmes de réécriture qui est fondée sur un codage complexe du problème de l'arrêt des machines de Turing ; j'ai simplifié et internalisé cette preuve en la réduisant à des problèmes indécidables connus des systèmes de réécriture : le problème de correspondance de Post et la terminaison des systèmes de réécriture monadiques ou semi-Thuiens [AR-14]. Cette réduction et le codage associé ont été repris par d'autres chercheurs pour d'autres preuves d'indécidabilité. mes travaux récents sur la normalisation forte des calculs des calculs de substitutions explicites [AR-16] et [AR-35] m'ont en quelque sorte ramené à cette problématique.

La surréduction. Examinant une problématique tout différente et une extension récente des systèmes de réécriture, j'ai étudié avec Pierre Réty, Claude Kirchner et Hélène Kirchner une méthode de résolution d'équations appelée *surréduction* qui généralise en quelque sorte la réécriture [CI-10].

La confluence. à la suite de travaux de Max Dauchet et Sophie Tison, je suis revenu sur le problème de la confluence. En effet, avec ces dernier et l'un de mes étudiants Thierry Heuillard nous avons proposé un algorithme de *décision de la confluence des systèmes de réécriture de termes clos* [CI-15,AR-10]. Ma contribution essentielle a été de trouver un langage adéquat à base de système de réécriture pour expliquer un algorithme complexe.

La complétion. La complétion est à la base d'une méthode très puissante de preuve par récurrence, appelée par jeu de mot l'induction sans induction. Cette preuve nécessitait de *tester la complétude*. En 1983, un de mes étudiants de DEA, Jean-Jacques Thiel, a proposé un algorithme très astucieux et très profond pour faire cela. Avec un autre étudiant Azzedine Lazrek nous l'avons étendu et implanté dans REVE [AR-7]. Hubert Comon a généralisé l'algorithme de Thiel en proposant une méthode intéressante pour l'aborder, basée sur des diséquations, ce sont des problèmes équationnels qui font intervenir la négation de l'égalité. Nous avons abordé systématiquement cette question, dans une communication avec Claude Kirchner [CI-16] et dans un article avec Hubert Comon [AR-8]. ce dernier article a été une source d'inspiration pour les recherches sur la résolution de contraintes symboliques.

La complexité. Dans un autre domaine, je me suis intéressé à l'évaluation de la complexité de l'algorithme de décomposition. Pour pouvoir l'étudier, je me suis limité, dans un premier temps, au cas monadique [CI-5]. Cela m'a amené à définir une structure de données, les *tournois avec clés répétées* [RI-5] dont Jean-Marc Steyaert et moi avons étudié quelques comportements dans [CI-7].

REVE

D'août 80 à août 82, j'ai fait un séjour post-doctoral au Massachusetts Institute of Technology invité par le professeur John Guttag, le spécialiste des types abstraits et le premier à avoir proposé un logiciel automatique de preuves dans ce cadre. C'est pendant cette période que j'ai effectué une partie de mes travaux sur les systèmes de réécriture de termes et notamment l'implantation du logiciel REVE, un environnement pour ces systèmes. REVE fut un véritable *laboratoire de réécriture* qui permettait de faire des preuves dans les spécifications algébriques de types abstraits [CI-6] mais aussi en algèbre.

Grâce à REVE, j'ai mené un certain nombre d'études expérimentales. Pour commencer, dans le domaine des spécifications algébriques, j'ai effectué quelques expériences encourageantes qui ouvraient des perspectives réelles pour REVE comme outil de certification. Mais c'est surtout en algèbre que j'ai obtenu les résultats les plus intéressants. En effet, j'ai pu découvrir, grâce aux ordres puissants de REVE, un nouveau système de réécriture pour décider la théorie des groupes, différent de celui classique qu'avait trouvé Knuth et Bendix et inconnu jusque là. Cela a permis à REVE de montrer le fait non trivial que l'unique axiome d'Higman et Neuman présente effectivement les groupes, ce qui constituait la première preuve par ordinateur de ce résultat. J'ai aussi pu mener à terme un exemple qui divergeait avec l'implantation de Knuth et Bendix. Cela a été présenté à la conférence on Automated Deduction en 1984 [CI-9].

Autour de REVE s'est créée une collaboration internationale qui a fait intervenir des chercheurs du MIT, du laboratoire de recherche de General Electric à Schenectady, de l'Université d'Illinois à Urbana-Champaign, de l'Université de l'État de New-York à Stony Brook. Cependant je considère qu'aujourd'hui l'aventure REVE est terminée. Cependant REVE a un successeur développé au MIT appelé LP ou LARCH PROVER qui a aussi de beau succès à son actif.

ORME

De récents progrès dans la présentation des algorithmes de complétion m'ont conduit à reconsidérer ma vision de la complétion et de son implantation. En effet, Leo Bachmair et Nachum Dershowitz ont proposé de décrire les procédures de complétion à l'aide un ensemble de règles de transitions. Leur but était de présenter les preuves de correction de façon plus élégante, j'ai pensé que ce qui était vrai pour les preuves devait être vrai pour les logiciels. Ainsi il est possible d'utiliser la même idée de règle de transition pour programmer élégamment et efficacement les procédures de complétion; cela permet de présenter très agréablement et très lisiblement le contrÃle. J'ai mis en œuvre cette idée en programmant moi-même un logiciel que j'ai appelé ORME. Bien sûr, les objectifs étant différents voire opposés, les mêmes règles de transitions ne peuvent servir à la fois pour les preuves de correction et pour la programmation. J'ai donc développé de nouvelles règles de transition en adoptant une démarche systématique de raffinement de ces règles pour montrer leur influence sur l'efficacité et expliquer mieux leur fonctionnement [CI-18]. Avec l'aide d'Houda

Chabbi, j'ai ensuite étendu cette approche à la complétion associative et commutative obtenant ainsi une procédure efficace et fiable [CI-20]. J'ai récemment ajouté l'ordre à base d'interprétations élémentaires [CI-23]. Bien que construit sans concession à la lisibilité, ORME est presque aussi rapide que les implantations classiques. Outre ses apports conceptuels et méthodologiques, ORME contient une boîte à outils sur laquelle d'autres logiciels peuvent être construits, il est distribué et utilisé. En 1993, un portage en CAML-Light a été fait et la version est actuellement distribuée [DL-3]. Mes doctorants et moi l'utilisons pour nos expériences de complétion ou de preuve de terminaison. Sébastien Briaïs élève à l'ENS l'a de sa propre initiative porté en OCAML.

La déduction automatique appliquée aux protocoles (90-00)

Ayant acquis une certaine expérience, j'ai voulu examiner comment les outils de démonstration automatique peuvent être mis en œuvre pratiquement pour prouver des propriétés de logiciels et de circuits matériels. En 90, ma vision était qu'il fallait regarder le long terme et le court et moyen terme. Plus précisément, je veux dire qu'il existait des outils de preuves très rigoureux et très sûrs, mais qui requerraient des connaissances pointues et une mise en œuvre moins immédiates, ils n'étaient pas à mon avis prêts à entrer dans le domaine des applications industrielles, mais j'étais convaincu qu'ils le seraient dans un futur plus ou moins proche (ce qui est le cas aujourd'hui). Il y avait d'autre part des outils qui semblaient aptes à prendre en compte presque immédiatement les applications industrielles, le tout était de savoir comment. Dans la première catégorie, se trouve COQ et le *Calcul des Constructions* et dans la seconde je place LP et le démonstrateur de Boyer et Moore. Concernant COQ, Joseph Rouyer a conduit sous ma direction la programmation d'un algorithme d'unification du premier ordre qui est à la fois une preuve de sa correction puisque le Calcul des Constructions ne fait pas de différence entre programme et preuve de la correction du programme [RI-10]. Concernant la deuxième approche, Boutheina Chetali a prouvé en LP un algorithme de division sans restauration [WK-1], un protocole de communication sur un canal non fiable [CI-28] et divers autres protocoles comme le contrôle d'un ascenseur. Stefan Krischer a, quant à lui, étudié la preuve de circuits sous divers aspects dont les machines à états finis et a obtenu des résultats extrêmement novateurs. Le groupe national CQFD que j'ai initialisé dans le cadre du GDR Programmation et dont Siva Anantaraman a pris la direction, étudie ces aspects.

Les calculs avec substitutions explicites (92-...)

Le λ -calcul est le modèle privilégié de la programmation fonctionnelle, mais aussi l'outil de la déduction automatique d'ordre supérieur et à ces titres nous intéresse au plus au point. Le problème de la substitution y revêt un caractère essentiel aussi bien pour l'implanteur que pour le théoricien, car c'est le mécanisme de calcul et de déduction fondamental. Il est traditionnellement

traité implicitement c'est-à-dire à l'extérieur du formalisme. Le calcul de substitutions explicites créé par Abadi, Cardelli, Curien, Hardin et Lévy suggère d'*internaliser* le traitement des substitutions proposant ainsi des systèmes de réécriture du premier ordre qui m'ont passionné d'emblée. J'ai, de fait, trouvé plusieurs systèmes [CI-24] alternatifs et plus simples dont l'un, le calcul $\lambda\nu$, est particulièrement élégant [RI-17]. De là, j'ai déduit des machines paresseuses à environnement que j'ai étudiées en collaboration d'étudiants doctorants sous les deux aspects du parallélisme et du partage de données.

Les calculs de substitutions explicites sont fondés sur les *indices* de de Bruijn ce qui les rend peu lisibles. Or de Bruijn dans son article fondateur introduit aussi ce qu'il appelle les *niveaux*. En les utilisant, on rend les substitutions explicites beaucoup plus lisibles. Nous avons donc défini avec Jocelyne Rouyer un calcul appelé $\lambda\chi$ [CI-25] qui se fonde sur ces idées. Ayant rencontré Nicolas de Bruijn à la conférence sur la déduction automatique, il m'a fait connaître ses travaux plus anciens et j'ai en particulier découvert que c'est lui le réel créateur du calcul des substitutions explicites et j'ai écrit une petite note présentant ce calcul dans nos notations actuelles [RI-19]. Un calcul de substitutions explicites appelé $\lambda - \tau$ avait été proposé sans que soit connu la terminaison de son calcul de substitutions. Frédéric Lang et moi avons montré par un contre-exemple que ce calcul pouvait conduire à des réécritures infinies [AR-17]. J'ai aussi écrit un article de présentation en français de l'ensemble de ces recherches [CN-7] ainsi qu'une présentation en anglais sous forme de types abstraits comme support d'une conférence invitée au workshop WADT [WK-4] et une autre qui rassemble mes travaux et ceux de mes étudiants Zino Benaïssa et Daniel Briaud, comme conférence invitée à la conférence APPIA-GULP-PROD'96 [CN-9].

Normalisation forte Contrairement à $\lambda\sigma$, $\lambda\nu$ préserve la forte normalisation de la β -réduction, la preuve donnée dans [RI-17] méritait d'être améliorée, ce que nous avons fait dans [AR-21]. Ces démonstrations assurent la normalisation forte des termes purs typés, c'est-à-dire sans substitutions, mais pas des autres termes typés. Daniel Dougherty (U. Wesleyan, Middletown, Connecticut, puis Worcester Polytechnic Institut, Massachusetts) et moi avons étudié divers systèmes de type pour garantir la normalisation faible et la normalisation forte des termes des calculs de substitutions explicites [AR-18, CI-29]. Dans ces travaux nous avons caractérisé les termes normalisables par réduction de tête et par réduction à l'extérieur à gauche, nous avons montré que tous les termes typés sont fortement normalisables, mais nous n'avons pas réussi à caractériser les termes fortement normalisables. Par un autre système de types et en collaboration avec un étudiant de l'ENS, Stéphane Lengrand nous avons finalement complètement caractérisé les termes fortement normalisables [CI-30]. Parallèlement Mariangiola Dezani et Steffen van Bakkel ont trouvé un système de type différent pour caractériser aussi les mêmes termes fortement normalisables, nous avons décidé d'écrire un article commun présentant les deux approches [AR-18].

Application à la sémantique des langages de programmation Avec Zino Benaïssa nous avons défini une structure de machine appelée *machine à triades* [RI-20]. Avec Kristoffer Rose (DIKU, ENS Lyon, IBM Yorktown Heights), nous avons prolongé ce travail en étudiant un calcul de substitutions explicites avec adresses qui permet d'étudier avec précision et rigueur le partage dans les implantations des langages fonctionnelles [CI-27]. Daniel Dougherty (U. Wesleyan, Middletown, Connecticut), Frédéric Lang, Luigi Liquori (ENS Lyon, INRIA Lorraine) et moi avons poursuivi cette étude en un calcul dit λObj^{a+} qui sert de cadre à une définition rigoureuse et générique de la sémantique opérationnelle des langages objets et qui prend en compte tous les concepts fondamentaux des langages à objets. [CI-28, RI-34].

L'interprétation du calcul des séquents classique (02-...)

Pierre-Louis Curien et Hugo Herbelin ont proposé un langage dit $\bar{\lambda}\mu\tilde{\mu}$ qui interprète le calcul des séquents classique. Silvia Ghilezan et moi y avons introduit un système de type avec intersection pour caractériser les termes fortement normalisables [CI-31] et avec Dan Dougherty [CI-32] un système de types avec intersection et union.

La Logique épistémique et jeux (03-...)

Je m'intéresse depuis plusieurs années à la logique épistémique ou logique de la connaissance qui règle les rapports entre agents et décrit aussi ce qu'est la connaissance commune. Cette recherche a pris un tour nouveau en 2004 avec deux activités de collaboration :

- séminaire sur la rationalité et la connaissance commune avec des collègues philosophes de l'ENS de Lettres et Sciences humaines de Lyon dont Sacha Bourgeois-Gironde,
- collaboration avec le Japanese Advanced Institute of Science and Technology (JAIST) sur la formulation de la théorie des jeux et les équilibres de Nash (René Vestergaard), cela donne lieu à la thèse en cotutelle de Stéphane Le Roux.

J'ai publié un compte-rendu de mes travaux dans *Annals of Mathematics and Artificial Intelligence*.

Modélisation biologique (06-...)

Ma nouvelle passion est la modélisation biologique, plus précisément les réseaux de régulation de gènes. Cette collaboration se fait avec René Vestergaard et Franck Delaplace de l'IBISC à Evry et a donné lieu à une publication à la conférence *Computational Methods in Systems Biology* [CI-35]. J'ai écrit un tutoriel [RI-36].

Le commerce électronique (95-01)

Comme nous l'avons dit, la preuve peut s'appliquer à des systèmes critiques comme ceux du commerce électronique. Dans le cadre du GIE DYADE entre Bull et l'INRIA je participe à l'opération VIP qui signifie *Verified Internet Protocols* et qui vise à certifier par preuves divers protocoles. Plus récemment, j'ai décidé de me consacrer à la preuve de transactions sécurisées décrites en JAVA. D'autre part, Barbara Heyd, qui a fait une thèse sous ma direction, a collaboré avec Pierre Crégut du CNET sur la preuve de spécifications écrites en UNITY et sur leurs applications à la vérification d'un protocole lié à ATM.

J'ai collaboré avec une entreprise lyonnaise sur le micro-paiement électronique, pour lequel notre équipe a obtenu un soutien de la part du Ministère de l'Industrie dans le cadre d'un projet OPPIDUM. Cette recherche a abouti à la conception d'un protocole de micro-paiement et à sa formalisation en COQ [WK-5].

EURECA

En 1978-1979, l'achèvement de ma thèse et sa mise en évidence de l'intérêt de la réécriture a coïncidé avec l'arrivée de Jean-Pierre Jouannaud. Lui et moi avons commencé à mettre sur pied une équipe de recherche dénommée EURÉCA qui peu à peu s'est forgée une bonne réputation internationale et a produit des résultats appréciés. Elle a d'ailleurs reçu en 1987 la médaille d'argent du CNRS. Euréca a participé à la création du GDR Programmation et Outils de l'Intelligence Artificielle et a publié ses résultats dans les grandes conférences internationales, ICALP, POPL, CADE, IJCAI, CAAP, LICS (en 95, 17 communications à des conférences et 15 articles dans des journaux). Nous avons mis sur pied une conférence internationale sur les *Techniques de la Réécriture et leurs Applications* qui en est à sa quatrième édition et compte parmi les grands événements du domaine, j'ai été président du comité de programme de la deuxième conférence qui a eu lieu à Bordeaux en mai 1987 et l'éditeur des actes [E-1] ainsi que le rédacteur en chef du numéro spécial de *Theoretical Computer Science* qui en a publié les meilleures communications [E-2]. Aujourd'hui EURÉCA est associée à l'INRIA. En 1992, elle s'est restructurée pour donner naissance à deux nouvelles équipes AMII et PROTHERO. En 1996, nous avons à nouveau entamé un processus de restructuration qui devrait aboutir dans le courant de l'année 1997 (une version provisoire du projet est disponible). En 1993, avec Ursula Martin professeur à l'Université de Saint Andrews en Écosse et Adam Cichon nous avons organisé la première édition d'un *workshop international sur la terminaison*; nous avons organisé la deuxième édition en 1995 à la Bresse dans les Vosges tandis que la troisième édition aura lieu près d'Utrecht. Les deux premières réunions ont rassemblé une soixantaine de chercheurs du monde entier.

PLUME

A mon arrivée à Lyon, j'ai monté une équipe appelée PLUME. Les deux principaux sujets de recherche sont les méthodes formelles et l'ordre supérieur. Cette équipe de recherche comporte trois permanents (dont deux maîtres de conférences, Jean Duprat et Daniel Hirschhoff) et trois doctorants et travaille sur la mobilité et la sécurité et sur les assistant de preuve comme COQ.

Autres aspects de ma recherche

Outre les points cités plus haut, d'autres aspects de l'informatique m'ont intéressé. Ainsi avec d'autres collègues j'ai écrit un livre sur la théorie des programmes [L-1], il est devenu l'ouvrage de base de l'enseignement d'informatique théorique dans les deuxièmes cycles des universités francophones (et australiennes !). Je me suis intéressé, d'autre part, à la construction des programmes [CI-2, CN-3, RI-4]. J'ai aussi épisodiquement collaboré à Médée ¹⁶ pour lequel j'ai proposé une sémantique [CN-1, CN-2]. J'ai présenté le langage CLU à la communauté de recherche, notamment par deux exposés aux journées *Langages Orientés Objets* de l'AFCEC [CN-5] et je persiste à penser que CLU est un langage extrêmement élégant et précurseur, c'est d'ailleurs pourquoi je me suis engagé dans la traduction du livre de Barbara Liskov et John Guttag un excellent manuel qui présente une méthodologie associée à ce langage [TR-1]. En 1989, j'ai préparé la candidature de John Backus comme Docteur Honoris Causa de l'Université Nancy 1 et j'ai organisé son séjour à Nancy lors de la remise du titre.

Au cours de ma carrière, j'ai été amené à faire des séminaires dans les laboratoires suivants (liste non exhaustive) :

- Bell Laboratories (Murray Hill, New Jersey) à l'invitation de Dave Mc Queen.
- MIT à l'invitation d'Albert Meyer.
- Université d'Illinois à Urbana-Champaign à l'invitation de Nachum Der-showitz et de Dave Plaisted.
- Louisiana Tech, à l'invitation de Dallas Lankford.
- Research and Development Laboratory de General Electric (à Schenectady New York) à l'invitation de Dave Musser.
- GTE Laboratories, (Waltham Massachusetts) à l'invitation d'Edward Johnson.
- Dortmund Universität (RFA) à l'invitation d'Harald Ganzinger.
- Swedish Institute for Computer Science à l'invitation de Philippe Mattieu.
- Institute for New Computer Generation (Tokyo, Japon).
- Carnegie Mellon University (Pittsburgh, Pennsylvanie, USA), à l'invitation de Jeannette Wing et Dana Scott.
- Institut d'informatique de la branche sibérienne de l'académie des sciences d'URSS (Akademgorodok, Russie, URSS), à l'invitation de Gregory Kucherov.

¹⁶MÉDÉE est le langage de la *Méthode Dédutive* chère à Claude Pair

- Institut de cybernétique V. Glushkov (Kiev, Ukraine, URSS), à l’invitation de Valentin Antimorov.
- Université de Strasbourg, Unité du CNRS Â«Fondements de la ScienceÂ», à l’invitation du Professeur Hervé Barreau, philosophe.
- Université de Wrocław (Pologne), à l’invitation de Leczek Pacholski.
- Université de Portland, à l’invitation de Sergio Antoy.
- Université de Stanford à l’invitation de Solomon Feferman.
- Boston University, à l’invitation de Wayne Snyder.
- Université Wesleyenne, (Middletown, Connecticut, USA) à l’invitation de Daniel Dougherty.
- Université de Provence, à l’invitation de Solange Coupet.
- Université de Novi-Sad, à l’invitation de Silvia Ghilezan.
- Worcester Polytechnic Institute, (Worcester Massachusetts) à l’invitation de Daniel Dougherty.

auxquels il faut ajouter des laboratoires français (INRIA Rocquencourt, IRISA, IRIT, LABRI, LIENS, LIP, LRI, ...).

J’ai de plus rendu visite aux laboratoires étrangers suivants, sur l’invitation de leurs chercheurs : IBM Yorktown Heights (W. Burge), Xerox Park (J. Horning), Université du Texas (R. Boyer et J Moore), SRI International (J. Goguen et J. Meseguer), ETL à Tsukuba (K. Futatsugi et T. Shimada). J’ai été invité pour des séjours d’un mois au SRI au cours de l’été 1985 et à l’ICOT au cours de l’été 1987.

Dans le cadre de la diffusion des connaissances scientifiques, j’ai enseigné à des écoles d’été, celle de l’AFCET [RI-2] en 1974, celle *Advanced Programming Technologies* à San Sebastian [CI-11] en septembre 1985, j’ai participé de nombreuses années aux DEA des universités de Nancy et de Lyon. J’ai rédigé avec Jean-Pierre Jouannaud divers articles de vulgarisation sur la réécriture, dans Enjeux [AN-1], TSI [AN-2], le bulletin de l’INRIA. Je participe chaque année à la Semaine de la Science à Lyon. J’ai été interviewé en 2003 par la chaîne de télévision lyonnaise TLM sur la sécurité des cartes bancaires et du commerce électronique.

Conférences invitées ¹⁷

J’ai été invité à donner des conférences au workshop international, *Combining Specification Methods*, à Nyborg au Danemark [RI-7] (mai 1984), au cinquième workshop *Semantics of Programming* à Bad-Honnef en Allemagne (mars 1985), au workshop *Combinatorial Algorithms on Algebraic Structures* à Otzenhausen en Allemagne (trois conférences, octobre 1985), au symposium international on *Trends in Computer Algebra*, organisée par IBM sur le calcul symbolique à Bad Neuenahr [CI-17] (mai 1987), au colloque on *Resolution of Equations in Algebraic Structures* organisé par le MCC et l’INRIA à Austin au Texas. (mai 87), à TAPSOFT [CI-18] (avril 87), à la conférence Algebraic and Logic Programming an 90 [CI-20] (septembre 90) et au symposium international *Mathematical Foundations of Computer Science* [CI-21] (septembre 91).

¹⁷Je ne cite que les conférences pour lesquelles j’ai rédigé un document écrit d’une dizaine de pages environ

J'ai donné une conférence invitée au workshop *Abstract Data Types* à Oslo en septembre 1995 en compagnie de Tony Hoare, Rod Burstall et Jim Horning et une conférence invitée à la conférence APPIA-GULP-PROD'96 [CI-30].