

Rapport INRIA 1994 — Programme 2

Parallélisme

PROJET PARA

3 mai 1995

PROJET PARA

Parallélisme

Localisation : *Rocquencourt*

Mots-clés : allocation de registre (7), architecture multiprocesseur (1), concurrence (1, 3), lambda-calcul (1, 5, 8), ordonnancement d'instructions (1, 7), parallélisme (1, 7), parallélisme asynchrone (1, 3), preuve de programme (1), preuve formelle (3), programmation fonctionnelle (1, 4, 5, 8), ramasse-miettes (1, 3), réécriture (6).

1 Composition de l'équipe

Responsable scientifique

Jean-Jacques Lévy, directeur de recherche

Responsable permanent

Georges Gonthier, chargé de recherche

Personnel INRIA

Robert Ehrlich, ingénieur de recherche

Luc Maranget, chargé de recherche

Chercheur post-doctorant

Eric Duquesne

Conseiller scientifique

Thérèse Hardin, professeur à Paris 6

Secrétariat (commun avec Coq et Cristal)

Ghislaine LeCorre

Sylvie Loubressac

Chercheurs doctorants

Damien Doligez, AMN, ENS-Ulm, de retour de service militaire à l'Ecole polytechnique, le 1-9-94,
Florent Guillaume, ENS-Ulm, depuis le 1-8-94 en service militaire à l'Ecole polytechnique,
Paul-André Melliès, boursier BFR, ENS-Ulm, depuis le 1-10-94 en service militaire à l'université libre d'Amsterdam,
Bruno Pagano, boursier MESR, depuis le 1-10-94,
Marc Pouzet, boursier INRIA, à Vérimag depuis le 1-10-94

Stagiaires

Henry Laulhère, du 15-4-94 au 1-8-94, Ecole polytechnique

2 Présentation du projet

Le projet Parallélisme consiste à étudier certains aspects de la programmation parallèle. De manière globale, l'évolution des ordinateurs vers des architectures parallèles nous semble être un des paris motivants de notre projet. En effet, la technologie permet aujourd'hui de construire facilement des structures multi-processeurs et la multiplication des processeurs permet un gain significatif de vitesse à technologie égale. Comme nous ne pouvons traiter tous les aspects du domaine, il nous semble par ailleurs que le parallélisme non massif (moins de 100 processeurs) est un problème intéressant et réaliste.

L'intérêt du groupe est à la fois porté vers la sémantique des langages et les systèmes de preuve, mais aussi vers l'implémentation de langages parallèles et à plus long terme vers la construction de systèmes matériels et d'exploitation parallèles. Pour ce faire, il nous semble que le projet doit être ouvert, c'est-à-dire maintenir des relations techniques suivies avec les autres projets théoriques ou pratiques de l'INRIA.

Les langages dont nous étudions les relations avec le parallélisme sont des langages fonctionnels pour trois raisons : 1) ils sont définis mathématiquement et donc plus simples et plus propres, 2) ils sont facilement disponibles dans le contexte de l'INRIA, 3) ils correspondent à la culture technique des personnes de notre groupe.

Pour mener une activité pratique, nous utilisons des machines multi-processeurs (un Sequent B8000 à dix processeurs et à mémoire partagée, un Encore MultiMax avec 14 processeurs, et une KSR à 72 processeurs).

Deux environnements ML parallèles ont été développés : un environnement ML standard en appel par valeur et un environnement ML paresseux. Un glâneur de cellules très performant a été conçu et est en cours de certification mécanique. Le système GAML a été complètement bootstrappé.

Quant à la théorie, c'est une activité importante de notre groupe en 1994. Elle se concentre sur les problèmes fins de l'implémentation des langages fonctionnels : partage de lambda-expressions, partage de réseaux de preuves dans la logique linéaire, théorèmes des développements finis et de normalisation forte, réécritures de graphes, généralisations des résultats précédents aux systèmes de réécriture abstraits, filtrage.

Nous regardons aussi les relations entre le parallélisme asynchrone et les langages fonctionnels, c'est-à-dire entre calcul de la communication (CCS et π -calcul de Milner) et le lambda calcul. Cela constitue un des buts essentiels de nos activités théoriques et pratiques futures.

Cette année est marquée par la soutenance de thèse de M. Pouzet, concluant son travail sur la compaction de code.

3 Actions de recherche

3.1 Récupération concurrente de mémoire et preuves mécaniques

Participants : Damien Doligez, Georges Gonthier

D. Doligez et G. Gonthier ont conçu, modélisé et prouvé correct un algorithme de récupération de mémoire concurrent et efficace qui permet de mettre en œuvre l'architecture décrite par D. Doligez et X. Leroy (projet CRISTAL) dans leur article de POPL 92. En effet, cette architecture permet de réaliser la récupération de mémoire de manière remarquablement efficace sur une machine multiprocesseurs, à condition de disposer d'un collecteur réellement concurrent, bien que pas nécessairement très rapide. Or, il s'est avéré qu'un tel collecteur n'existait pas ; tous les auteurs s'étant attaqués dans la foulée de Dijkstra et Lamport à ce problème ont sous-estimé la synchronisation extrêmement délicate nécessaire à sa solution : leur solution, si elle n'était pas fautive, impliquait un surcroît de synchronisation qui détruisait la concurrence de l'algorithme.

C'est par le biais d'un modèle rigoureux et détaillé du problème que D. Doligez et G. Gonthier ont pu mettre au point un algorithme de récupération de mémoire parfaitement concurrent ; bon nombre des détails les plus délicats de l'algorithme ont été fixés par la preuve formelle de sa correction. Ces travaux ont fait l'objet d'une présentation à POPL 94, à Portland, Oregon (USA), puis de communications à DEC-SRC et à Xerox-PARC.

D. Doligez et G. Gonthier se sont alors demandé s'il était possible de mener une preuve sur ordinateur de leur algorithme. Après quelques tâtonnements, cette preuve est actuellement en cours de développement dans le système TLP (Guttag, Horning et Engberg) avec le formalisme des TLA (*Temporal Logic of Actions*) de Lamport. Pour cette preuve extrêmement combinatoire, il suffit d'un système de démonstration assistée avec un interface simple et robuste comme le mode Emacs de TLP, un bon préprocesseur pour une logique modale élémentaire (TLA), et un système de réécritures équationnelles (LP). Si ce système n'a pas la fonctionnalité puissante de démonstrateurs comme Coq ou HOL, il est sûr que la preuve menée à terme constituera un bon exemple de comparaison, sur une preuve non académique. Actuellement, elle contient 600 lignes et 400 règles de TLP pour démontrer 4 des 60 cas de l'article POPL'94, et a demandé 5 mois. On doit noter qu'il n'est pas évident de mener à terme une preuve mécanique de telle taille.

D. Doligez a arrêté cette activité pour rédiger sa thèse sur les systèmes d'exécution et de récupération mémoire, éventuellement concurrents.

3.2 Langages de programmation fonctionnels

Participants : Damien Doligez, Luc Maranget

L. Maranget a poursuivi le développement du système fonctionnel paresseux GAML. Le système GAML a été porté sur la machine KSR (72 processeurs) de l'INRIA. Le bootstrap de la version séquentielle est achevé. La version parallèle est au stade final de développement et de l'analyse de performances. Le compilateur GAML étant écrit en GAML, on voit poindre la possibilité de paralléliser le compilateur lui-même et de procéder à un bootstrap complet de GAML parallèle.

L. Maranget a écrit un article sur le filtrage avec priorité de ML. Cet article est la conclusion de deux années de travail sur la définition et la compilation du filtrage en ML. Il décrit exhaustivement et prouve la

correction des deux algorithmes de compilation du filtrage paresseux que L. Maranget a introduits et préalablement intégrés dans le compilateur GAML. Cet article a été soumis au Journal of Functional Programming, début septembre [13].

L. Maranget a également modifié la compilation du filtrage dans le système Caml-Light du projet CRISTAL. Les modifications introduites permettent maintenant au compilateur Caml-Light de détecter rigoureusement et statiquement les clauses de filtrage inutiles et les expressions de filtrage non exhaustives.

D. Doligez et F. Rouaix (projet CRISTAL) ont écrit un profileur pour Caml-Light, qui permet de compter les appels de fonctions, les branches de tests et de filtrages, etc. Le profileur est un outil intéressant pour mettre au point et améliorer les performances des programmes Caml-Light.

D. Doligez a apporté des améliorations importantes au GC de Caml Light (projet CRISTAL) pour réduire la fragmentation dans la liste libre des blocs alloués. Ceci a permis de gagner un facteur 3 en vitesse pour le système Coq (projet COQ).

Toutes les modifications menées dans notre projet seront intégrées à la prochaine version de Caml-light qui sortira en début 95.

D. Doligez et L. Maranget ont encadré Henri Laulhère, stagiaire de l'Ecole polytechnique, qui a porté le système Caml-light parallèle sur KSR.

3.3 Substitutions explicites

Participants: Thérèse Hardin, Jean-Jacques Lévy, Paul-André Melliès

T. Hardin a défini, en collaboration avec G. Dowek (projet COQ) et C. Kirchner (projet PROTHEO), un algorithme d'unification dans le $\lambda\sigma$ -calcul et a montré que cet algorithme peut être utilisé pour effectuer l'unification d'ordre supérieur (papier soumis à une conférence). En effet, la méthode utilisée dans la thèse de Huet (1977) introduit des β -radicaux arbitraires pour désigner en fait de nouvelles substitutions de variables. Dans le formalisme des substitutions explicites, une telle description est plus naturelle, puisque les substitutions, et non les β -radicaux, sont les éléments de base. Ainsi, on peut ramener l'unification d'ordre fonctionnel élevé à des problèmes de résolution d'équations dans la théorie des

substitutions explicites, sans avoir à se soucier des problèmes de portée des variables liées, introduits traditionnellement avec la β -conversion. La présentation de l'algorithme d'unification devient plus modulaire, évite les étapes inutiles et est plus proche de l'implémentation.

T. Hardin et J.-J. Lévy, en collaboration avec P.-L. Curien (Laboratoire d'Informatique de l'ENS) ont fait les dernières corrections à leur article sur la confluence des différents formalismes avec substitution explicites, accepté pour JACM.

P.-A. Melliès, dans le cadre de sa rédaction de thèse, a trouvé accidentellement un contre-exemple à la normalisation forte de la plupart des formalismes de substitutions explicites typés. Ce résultat était inattendu, puisque diverses personnes essayaient de démontrer le contraire. Philosophiquement, cela veut dire qu'un calcul «super-paresseux» peut boucler. Plus exactement, il apparaît comme très difficile d'obtenir un calcul de substitutions à la fois confluent et fortement normalisable, les seuls exemples fortement normalisables connus n'étant confluents que sur les termes clos. Ce résultat sera présenté à TLCA'95, à Edimbourg en avril 95.

Les rapports entre le $\lambda\sigma$ -calcul et la logique linéaire ont été étudiés par E. Duquesne, notamment au cours d'une série d'exposés à Paris 6 dans le groupe de travail de T. Hardin.

T. Hardin et L. Maranget ont encadré B. Pagano, dans le cadre de son stage de DEA, qui a exprimé les stratégies mises en œuvre par les machines abstraites comme les machines SECD, CAM et de Krivine à l'aide du calcul faible des substitutions explicites.

3.4 Systèmes de réductions abstraites

Participant : Paul-André Melliès

P.-A. Melliès travaille sur la terminaison abstraite en poursuivant un résultat obtenu l'année dernière pour le théorème des développements finis. La preuve qu'il avait alors proposée utilisait deux relations sur les radicaux: l'ordre habituel d'emboîtement des radicaux, la relation de deux radicaux dont un peut substituer sous l'autre, c'est-à-dire qui lie une variable que le second contient. Quatre axiomes sur ces deux relations permettent de décrire quels mécanismes vérifient l'ensemble des calculs d'ordre supérieur, qu'ils soient Church-Rosser ou non (systèmes à

réductions combinatoires de Klop dont certains sont non orthogonaux), qu'ils soient des réductions de termes ou de classes associatives commutatives (comme pour le γ -calcul de Boudol), ou bien de graphes. En plus de la description, cette construction fournit une preuve simple, là où on avait auparavant des preuves complexes (voir la thèse de Klop) ou même intrinsèquement incassables comme, par exemple, dans le cas des réductions combinatoires non orthogonales.

La réussite de cette méthode a poussé P.-A. Melliès à continuer pour obtenir une preuve combinatoire de terminaison forte du λ -calcul simplement typé. Il a fallu généraliser le système à un nombre quelconque de relations et montrer, au prix d'une preuve difficile, qu'un tel système aux axiomes élargis normalise fortement.

La seconde étape a été de prouver que le λ -calcul simplement typé vérifie lui-même ces axiomes. Cette étape est importante car, en plus d'apporter une meilleure compréhension dans le λ -calcul des mécanismes de terminaison, elle permet de remettre en question ce dont nous disposons pour le traiter. Cette remise en question est devenue nécessaire avec les travaux d'Asperti et de Laneve sur l'optimalité dans les systèmes à réductions combinatoires, qui nécessitent un emploi compliqué et non abstrait par manque d'outils descriptifs. C'est cette même absence qui rend difficile la preuve d'adéquation du λ -calcul simplement typé aux axiomes généralisés. Une notion nouvelle, celle de *tour*, a été introduite pour cette preuve. Certains rapports semblent se dessiner avec la théorie de l'optimalité dans le λ -calcul.

P.-A. Melliès s'est intéressé en dernier point aux problèmes de standardisation que le travail qu'il avait présenté à LICS'92 avec G. Gonthier et J.-J. Lévy avait laissés inachevés. La question est simple: comment obtenir la propriété de standardisation sur les graphes, et cela sans passer par la sémantique sous-jacente des arbres? On obtient alors une classification des systèmes de réécriture de graphe selon qu'ils pratiquent une duplication à tout moment (cas le plus simple) ou bien seulement lorsqu'il le faut (cas des graphes acycliques et du calcul optimal sur des termes du premier ordre). Cette distinction est primordiale car elle induit des traitements de la standardisation, et donc de l'intentionnalité du calcul, complètement opposés. On obtient la propriété de standardisation sur les graphes avec duplication permanente par un affaiblissement des axiomes.

3.5 Compaction des langages fonctionnels

Participant : Marc Pouzet

Marc Pouzet s'est intéressé à la compilation d'un langage fonctionnel pour des architectures à parallélisme fin, de type VLIW ou super-scalaires. Ces architectures sont capables d'exécuter plusieurs instructions élémentaires en parallèle. La compilation pour ce type d'architecture, ou *compaction*, consiste donc à rechercher le parallélisme entre les instructions élémentaires du programme pour construire des séquences d'instructions indépendantes qui seront exécutées en parallèle.

Cette année a été consacrée à la rédaction de sa thèse[1]. L'ambition de celle-ci est de montrer comment la compaction peut être décrite sur un langage fonctionnel simple par un ensemble de transformations de programmes. Pour cela, le langage fonctionnel utilisé est d'abord décrit, ainsi qu'une sémantique opérationnelle de son exécution sur une machine parallèle de type VLIW ou super-scalaire. Une équivalence dite «structurelle» mettant en relation les programmes ayant «les mêmes dépendances» est donnée ainsi qu'une procédure de décision partielle. Suit la description de la compaction globale (parallélisme exploité dans les sections sans récursions) puis celle du pipeline logiciel (compaction étendue aux programmes quelconques). Ainsi, le principe du «pipeline logiciel», habituellement appliqué aux boucles est étendu aux fonctions récursives. Il est obtenu par la compaction globale et le dépliage contrôlé des fonctions et fournit un nouveau programme dont la compaction ne peut plus être améliorée par dépliage. Diverses méthodes permettant de l'obtenir sont proposées.

Marc Pouzet a fait un exposé à la conférence internationale *Applications in Parallel and Distributed Computing* qui a eu lieu à Caracas en avril 1994 [9]. Il a soutenu sa thèse de Doctorat de l'Université Paris 7 le 14 octobre 1994.

T. Hardin a également travaillé, en collaboration avec M. Pouzet, sur une sémantique des λ -termes, intervenant dans la parallélisation automatique (papier en préparation).

3.6 Partage dans le lambda-calcul et la logique linéaire

Participants : Eric Duquesne, Georges Gonthier, Jean-Jacques Lévy

G. Gonthier est toujours en train d'écrire avec M. Abadi et J.-J. Lévy une version plus complète des articles présentés à POPL'92 et LICS 92 sur les réductions optimales dans le lambda-calcul et dans la logique linéaire.

E. Duquesne a travaillé dans la sémantique opérationnelle et dénotationnelle de la logique linéaire (LL). Il a étendu la définition des réseaux de preuves PN_0 de LL multiplicative aux réseaux de LL multiplicative et exponentielle en termes de graphes. L'étude se limite au cas intuitionniste avec la traduction PN du λ -calcul. Après la définition de l'interprétation de PN par un espace cohérent [3] et conséquemment après la définition nouvelle du lien *pourquoi-pas* avec indices de boîtes, la question de la définition des réseaux de PN comme graphes par oubli des indices de boîtes est résolue pour les graphes sans coupures exponentielles [4]. Le critère de correction obtenu est linéaire en la taille du graphe.

L'ensemble des réseaux de preuves sans coupures exponentielles de PN est en bijection avec le λ -calcul, l'élimination des coupures étant la β -réduction. Par conséquent, l'ensemble PN tout entier constitue une extension du λ -calcul, il consiste en une explicitation du calcul de la β -réduction et de la substitution; la recherche d'une définition en termes de graphes de ces réseaux conduit à travaux sur la substitution explicite, et le $\lambda\sigma$ -calcul. Une conséquence de cette définition est une nouvelle caractérisation du typage principal des λ -termes pour lequel un typage principal n'est rien d'autre qu'un réseau de preuves sans coupures (cf. [12]).

Enfin, E. Duquesne a considéré la géométrie de l'interaction pour formaliser la sémantique opérationnelle des réseaux de preuves de la logique linéaire et des graphes de partage, permettant ainsi d'en donner une nouvelle interprétation. Cela donne une définition originale du typage principal d'un réseau (traduit d'un λ -terme). On peut ainsi produire un algorithme dans l'esprit de la sémantique des contextes de Gonthier ou de la géométrie de l'interaction pour déterminer le typage principal cf. [12].

4 Actions industrielles

4.1 Projet Esprit CONFER

Dans le cadre des projets Esprit *Basic Research* (BRA), notre projet est le site coordinateur du contrat 6454 CONFER (*Concurrency and Functions : Evaluation and Reduction*). Ce projet regroupe le CWI (Klop), l'ECRC-Munich (Thomsen), l'université d'Edimbourg (Milner), l'École Normale Supérieure (Curien), l'Imperial College (Abramsky), l'Inria-Sophia-Antipolis (Boudol), l'université de Pise (Montanari) et le SICS-Stockolm (Parrow).

Le thème de ce projet est de regrouper les efforts sur la concurrence et le λ -calcul pour aboutir à l'ambitieux projet d'un calcul unifié. Quatre principaux axes en font partie : les modèles de calcul fondamentaux, les calculs plus spécifiques (π -calcul de Milner, relations entre concurrence et fonctionnalité étudiées principalement par Boudol et Sangiorgi), la logique et la concurrence d'ordre supérieur (Abramsky), les langages de programmation.

Dans cette dernière activité, l'ECRC-Munich développe le langage Facile qui offre la concurrence d'ordre fonctionnel supérieur et la distribution sur réseau. Un autre langage PICT est développé à Edimbourg, puis à Cambridge, et en partie à l'Inria par D. Rémy (projet CRISTAL). Notre projet Esprit aboutira à une meilleure compréhension des modèles de calculs distribués d'ordre supérieur, et à la définition de langages pour processus mobiles. [5]

5 Actions nationales et internationales

LICS'94 J.-J. Lévy est allé à Imperial College (Londres) les 20-21 février pour le comité de programme de la conférence LICS.

Séminaire Para-Formel Notre groupe a un séminaire commun avec les projets COQ et CRISTAL de Rocquencourt.

Séminaire d'informatique de l'X J.-J. Lévy participe à l'organisation du séminaire mensuel d'informatique de l'École polytechnique qui a reçu en 1994 MM. Kahn, Smolka, Cohen, Longo, Viennot, Diaz, Common, Karlin, Huet, et Toueg.

Séminaire de programmation fonctionnelle L. Maranget a animé un séminaire d'introduction aux langages de programmation fonctionnels le 3 mai, devant l'équipe Langage, Informatique, Logique et Cognition du Centre d'Analyse Mathématique et Sociale, une unité mixte dépendant du CNRS de l'École des Hautes Etudes en Sciences Sociales et de l'Université Paris 4.

Comité TSI T. Hardin est depuis janvier 1994 membre du comité de rédaction de TSI.

6 Diffusion des résultats

6.1 Diffusion de produits

Le système Caml est distribué par le projet CRISTAL en *ftp* anonyme. Son glaneur de cellules a été écrit par D. Doligez. Le système Gaml (paresseux) est toujours expérimental.

6.2 Actions d'enseignement

6.2.1 Enseignement universitaire

Ecole polytechnique J.-J. Lévy est professeur à l'École polytechnique. Il est responsable du cours « Algorithmes et Programmation » de tronc commun de première année (30 enseignants, 420 élèves, rédaction du polycopié disponible sur le Web [10], cours en amphitheâtre et petites classes, projets informatique). Il organise ce cours toutes les années paires en alternance avec R. Cori. J.-J. Lévy participe aussi au groupe de refonte du programme d'informatique en classes préparatoires.

G. Gonthier est Maître de Conférence à l'École polytechnique. Il dispense le cours de tronc commun d'informatique à deux petites classes de septembre à décembre.

Université Paris 6 T. Hardin est professeur à l'Université Paris 6. Elle y assure le cours de programmation et structures de données dans la Maîtrise de Sciences et Techniques d'informatique et y encadre plusieurs stages en entreprise. Elle est également chargée du cours de licence sur les fondements de la programmation. Elle effectue aussi le cours de λ -calcul du tronc commun

du D.E.A. d'Informatique Théorique, Calcul et Programmation et y enseigne, dans le cadre d'une option, la réécriture, les $\lambda\sigma$ -calculs et la programmation fonctionnelle.

Elle a animé à l'Université Pierre et Marie Curie un groupe de travail sur la logique linéaire, avec la collaboration d'E. Duquesne.

Ecole Nationale des Ponts et Chaussées L. Maranget a enseigné à l'École nationale des ponts et chaussées en tant que maître de conférence vacataire, dans le cadre du cours de première année « Informatique fondamentale », de janvier à mars.

DEA X-ENS-INRIA G. Gonthier participe à l'enseignement du DEA IMA et donne un cours sur les méthodes de vérification des processus concurrents.

6.2.2 Autres actions d'enseignement

Séminaire Coq au CNAM T. Hardin a organisé, en collaboration avec V. Donzeau-Gouge Viguié, deux journées de formation à Coq et CTCoq, animée par des membres des projets COQ (G. Dowek, A. Saïbi, B. Werner) et CROAP (Y. Bertot). Ces journées se sont déroulées en juin et octobre dans les locaux du CNAM (50 participants).

Cours INRIA R. Ehrlich a donné pour la 4ème fois un cours sur le noyau Unix dans le cadre des cours de formation interne à l'INRIA.

6.2.3 Jurys de thèse

T. Hardin a été membre du jury de soutenance de la thèse d'Antonio Bucciarelli (Paris 11). J.-J. Lévy a été membre du jury de la thèse d'habilitation de Roberto Amadio (Nancy) et de Marc Pouzet (Paris 7).

6.3 Participation aux manifestations

Workshops CONFER

Deux réunions dans le cadre du projet Esprit CONFER : quatrième et cinquième workshops du projet Esprit CONFER à l'ECRC à Munich, du 18 au 20 avril, et à Imperial College à

Londres, du 4 au 7 octobre. Y ont participé D. Doligez, E. Duquesne, G. Gonthier, F. Guillaume, J.-J. Lévy, L. Maranget, P.-A. Melliès.

Conférence IFIP ICAPDC

M. Pouzet a fait un exposé à la conférence *Applications in Parallel and Distributed Computing* à Caracas, 18-22 avril 1994 [9]

POPL'94

D. Doligez a exposé un article, écrit avec G. Gonthier [6], à la conférence POPL'94 de Portland, Oregon, en janvier. D. Doligez et G. Gonthier ont rendu visite à John Lamping (réductions optimales) et à Hans Bøehm (récupération mémoire) au centre de recherche Xerox-PARC et à Martin Abadi (réductions optimales) à DEC-SRC à Palo Alto.

LICS'94

E. Duquesne, G. Gonthier, T. Hardin, J.-J. Lévy, L. Maranget et P.-A. Melliès ont assisté à la conférence LICS, qui se tenait au CNAM, à Paris, les 4-7 juillet 1994.

6.4 Activités extérieures

- L. Maranget s'est rendu à l'Université de Norwich (Royaume Uni) du 15 au 20 mai. Il y a principalement travaillé avec Z. Khasidashvili.
- F. Guillaume a participé à la conférence Compiler Construction à Edimbourg, où était présenté un article co-écrit avec Lal George et John Reppy à AT&T Bell laboratories [7].
- T. Hardin a effectué un court séjour à l'INRIA-Lorraine, sur invitation de C. Kirchner, pour discuter de l'utilisation des $\lambda\sigma$ -calculs dans l'unification d'ordre supérieur. Elle y a fait un exposé sur les $\lambda\sigma$ -calculs.
- J.-J. Lévy a participé au jury CR de l'INRIA-Lorraine pour la deuxième année consécutive les 10 et 17 juin à Nancy.

7 Publications

Thèses

- [1] M. POUZET, *Compaction des Langages Fonctionnels*, Thèse de doctorat, Paris 7, octobre 1994.

Articles et chapitres de livre

- [2] P.-L. CURIEN, T. HARDIN, J.-J. LÉVY, «Confluence properties of Weak and Strong Calculi of Explicit Substitutions», *Journal of the ACM*, A paraître.
- [3] E. DUQUESNE, J. V. DE WIELE, «Modèles cohérents des réseaux purs PN», *Archive for Mathematical Logic* 33, 1994, p. 131–158.
- [4] E. DUQUESNE, J. V. DE WIELE, «PN as graphs», *Archive for Mathematical Logic*, 1994.
- [5] J.-J. LÉVY, B. THOMSEN, L. LETH, A. GIACALONE, «First Year Report for Esprit Basic Research Action 6454-CONFER – CONCURRENCY and Functions: Evaluation and Reduction», *Bulletin of EATCS*, 52, 1994, p. 164–174.

Communications à des congrès, colloques, etc.

- [6] D. DOLIGEZ, G. GONTHIER, «Portable, Unobtrusive Garbage Collection for Multiprocessor Systems», in: *Twenty-First Annual ACM Symposium on Principles of Programming Languages*, ACM, Portland, Oregon, janvier 1994.
- [7] L. GEORGE, F. GUILLAUME, J. REPPY, «A Portable and Optimizing Back End for the SML/NJ Compiler», in: *Compiler Construction*, Edimbourg, mars 1994.
- [8] P.-A. MELLIÈS, «Typed λ -calculi with explicit substitutions may not terminate», in: *Second conference on Types in the Lambda Calculus*, Edimbourg, 1995.
- [9] M. POUZET, «Fine Grain Parallelization of Functional Programs for VLIW and Super-sclalars Architectures», in: *IFIP International Conference on Applications in Parallel and Distributed Computing*, Caracas, avril 1994.

Cours polycopiés

- [10] R. CORI, J.-J. LÉVY, *Algorithmes et Programmation*, Ecole polytechnique, octobre 1994, <http://sil.polytechnique.fr:5035/poly>.

Rapports de recherche et publications internes

- [11] G. DOWEK, T. HARDIN, C. KIRCHNER, «Higher-order unification in $\lambda\sigma$ -calculus», *rapport de recherche*, INRIA, Rocquencourt, octobre 1994.
- [12] E. DUQUESNE, J. V. DE WIELE, «A new intrinsic characterization of the principal type schemes», *rapport de recherche*, INRIA, Rocquencourt, novembre 1994.
- [13] L. MARANGET, «Two Techniques for Compiling Lazy Pattern Matching», *rapport de recherche*, INRIA, Rocquencourt, septembre 1994.

8 Abstract

The PARA project studies interactions between concurrency and functional programming. This year, we worked on a mechanized proof of a concurrent garbage collector designed by us in 1993, in pattern matching and optimisation in functional languages, and in the theory of rewriting, with a counterexample to the strong normalisation of typed explicit substitutions.

Table des matières

1	Composition de l'équipe	1
2	Présentation du projet	2
3	Actions de recherche	3
3.1	Récupération concurrente de mémoire et preuves mécaniques	3
3.2	Langages de programmation fonctionnels	4
3.3	Substitutions explicites	5
3.4	Systèmes de réductions abstraites	6
3.5	Compaction des langages fonctionnels	8
3.6	Partage dans le lambda-calcul et la logique linéaire	8
4	Actions industrielles	10
4.1	Projet Esprit CONFER	10
5	Actions nationales et internationales	10
6	Diffusion des résultats	11
6.1	Diffusion de produits	11
6.2	Actions d'enseignement	11
6.2.1	Enseignement universitaire	11
6.2.2	Autres actions d'enseignement	12
6.2.3	Jurys de thèse	12
6.3	Participation aux manifestations	12
6.4	Activités extérieures	13
7	Publications	14
8	Abstract	15