INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Projet LEMME

Logiciels et Mathématiques

Sophia Antipolis

THÈME 2A

Table des matières

1. Composition de l'équipe

Responsable scientifique
Loïc Pottier [Chargé de recherche INRIA]

Responsable permanent
Yves Bertot [Directeur de recherche INRIA, HDR]

Assistante de projet
Nathalie Bellesso

Personnel Inria
Gilles Barthe [Chargé de recherche INRIA]
Janet Bertot [Ingénieur de Recherche INRIA, équipe DREAM, depuis juin]
Marieke Huisman [Chargée de recherche INRIA]
Francis Montagnac [Ingénieur de Recherche INRIA, jusqu'en octobre 2002]
Laurence Rideau [Chargée de recherche INRIA]
Laurent Théry [Chargé de recherche INRIA, en détachement au 1er Décembre 2002]

Fonctionnaire en délégation
Philippe Audebaud [Maître de conférence, ENS Lyon]

Fonctionnaire en détachement
Frédérique Guilhot [Professeur agrégée, académie de Nice, depuis septembre]

Chercheurs post-doctorants
Venanzio Capretta [bourse CEE]
Pierre Courtieu [bourse Verificard]
Sorin Stratulat [jusqu'en septembre, bourse Verificard]
Leonor Prensa [depuis septembre, bourse Profundis]
Christoph Sprenger [depuis octobre, bourse ERCIM]

Collaborateurs extérieurs
André Hirschowitz [Professeur UNSA, Laboratoire J.A. Dieudonné]
Roger Marlin [Professeur UNSA, Laboratoire J.A. Dieudonné]
Monica Nesi [Maître de Conférences, Université de L'Aquila, Italie]

Ingénieur en poste d'accueil-jeune
Ahmed Amerkad [jusqu'en juin]

Chercheurs doctorants
Antonia Balaa [ATER UNSA]
Néstor Cataño [bourse INRIA]
Laurent Chicli [ATER UNSA]
Kuntal Das Barman [bourse INRIA]
Guillaume Dufay [bourse MESR]
Kwong-Cheong Wong [bourse INRIA, jusqu'en février]
Hanane Naciri [bourse MESR]
Nicolas Magaud [bourse MESR]
Tamara Rezk [3 mois]
Simão Melo de Sousa [Boursier gouvernement Portuguais]
Kerry Trentelman [ANU, Canberra, jusqu'en mars]

Stagiaires
Assia Mahboubi [ENS Lyon août]
Florence Martel [CNAM, 6 mois]
Mélanie Vaillant [Maîtrise Math-Info, Nice, 6 semaines]

Sophie Boulmé [ESINSA, 2 mois]
David Baccialon [Maîtrise Math-Info, Nice, 6 semaines]
Farid Latrech [Maîtrise Math-Info, Nice, 6 semaines]

2. Présentation et objectifs généraux

Les recherches du projet Lemme ont pour but de diffuser les méthodes formelles dans la construction de logiciels, en particulier pour le calcul scientifique. Tout en y contribuant, le projet s'appuie sur les résultats de domaines liés aux méthodes formelles, comme la théorie des types, la sémantique des langages de programmation, l'algorithmique mathématique, le calcul formel, l'arithmétique des ordinateurs. Ses résultats sont confrontés à des applications dans le cadre des projets de recherche européens (mathématiques sur le Web, cartes à puces) ou par des collaborations avec d'autres équipes de recherche (arithmétique des ordinateurs, enseignement des mathématiques, calcul formel, sémantique des langages de programmation).

Le projet suit quatre directions de recherche principales :

Environnements de preuves, pour des utilisateurs ingénieurs, enseignants, chercheurs et étudiants.
Formalisation des mathématiques et théorie des types, pour décrire formellement les objets de base de la connaissance scientifique.
Certification d'algorithmes, pour développer des outils permettant de construire des implémentations efficaes et sûres d'algorithmes à partir de leurs spécifications et de leurs preuves de correction.
Formalisation de la sémantique de langages de programmation, avec un accent sur les programmes Java.

Notre outil privilégié est le système Coq, que nous utilisons intensivement. En collaboration avec le projet Logical, nous contribuons aussi à en développer certaines fonctionnalités. À noter qu'un livre sur Coq, écrit par Yves Bertot et Pierre Casteran, est en préparation.

3. Fondements scientifiques

3.1. Environnements de preuves

Mots clés : *preuve*, *environnement*, *interface homme-machine*, *Coq*.

Le but de ce thème est d'étudier les outils mécaniques de recherche et de vérification de preuves pour faciliter leur utilisation par des ingénieurs et des mathématiciens dans la production de logiciels et de théories mathématiques formelles.

3.2. Théorie des types et formalisation de théories mathématiques

Mots clés : *formalisation*, *mathématiques*, *théorie des types*, *Coq*.

Les buts de ce thème sont de développer la théorie des types, et d'étudier comment des théories mathématiques où interviennent de nombreux types d'objets peuvent être représentées dans le calcul des constructions inductives (CCI en abrégé), de façon à être aussi lisibles et utilisables que possible par quelqu'un qui en a une connaissance scolaire ou académique.

3.3. Implémentations certifiées d'algorithmes de calcul scientifique

Mots clés : *algorithme*, *certification*, *Coq*.

Pour obtenir des programmes certifiés, nous proposons une approche inverse de celle généralement utilisée : plutôt que de chercher à prouver des propriétés d'un programme existant (en formalisant sa sémantique), nous proposons de produire des programmes dont la correction découle de celle de leur processus de création.

3.4. Sémantique des langages de programmation

Mots clés : *sémantique*, *langages*, *programmation*, *Java*, *JavaCard*, *Coq*.

Les algorithmes intervenant dans l'implantation des langages de programmation font également partie de notre champ d'investigation. Pour ces algorithmes, on se repose généralement sur la description sémantique d'un langage, et les propriétés que l'on cherche à établir pour un algorithme sont soit qu'il préserve la sémantique des programmes (s'il s'agit d'un algorithme de transformation ou d'optimisation) soit que les programmes qu'il produit sont exempts de certains comportements indésirables (s'il s'agit d'un compilateur ou d'un vérificateur de programmes). Pour classifier ce type d'algorithmes et de vérifications, nous parlons de preuves en sémantique des langages de programmation.

4. Domaines d'application

4.1. Cartes à puces

La nouvelle génération de cartes à puces contient typiquement un microprocesseur et une mémoire (mais avec des limitations sur les calculs et le stockage). Elles sont souvent utilisées pour stocker des données sensibles, d'où le problème de leur sûreté. Ce qui en fait un domaine d'application des méthodes formelles.

Pour programmer des cartes à puces, on utilise des langages de programmation de haut niveau comme JavaCard (un dialecte de Java). Pour raisonner sur la correction des cartes à puces, on doit formaliser à la fois la plate-forme dans le standard JavaCard et le langage JavaCard lui-même.

Dans le projet Lemme, nous avons étudié des preuves de correction pour des vérificateurs de bytecode, en nous concentrant sur l'initialisation des objets et sur la sûreté des types. Notre travail actuel concerne le développement d'outils facilitant ces preuves, ainsi que des études de cas pour comprendre comment établir la correction d'applications de cartes à puces.

Ce travail est fait avec en collaboration avec le constructeur de cartes à puces Gemplus et dans le contexte du projet européen IST Verificard.

4.2. Algorithmes certifiés

Pour certaines applications, il est nécessaire d'avoir des logiciels sans erreurs. Un moyen d'atteindre ce but est de développer en parallèle avec un programme sa preuve de correction. Dans le projet Lemme, nous nous intéressons à obtenir des programmes certifiés effectuant des calculs scientifiques, en suivant une méthode qui consiste à partir non pas du programme mais de sa description algorithmique abstraite. Prouver d'abord la correction d'un algorithme a l'avantage principal que l'on se concentre alors sur les aspect mathématiques les plus profonds. L'étape suivante consiste à dériver une implémentation efficace de l'algorithme, plus ou moins automatiquement, mais suivant un procédé sûr. Nous avons expérimenté cette approche sur des exemples d'arithmétique, de calcul formel, de géométrie algorithmique.

4.3. Web, MathML, XML

Notre travail autour d'XML et MathML, dans le contexte de Figue, devrait nous permettre de partager des preuves (scripts Coq ou des arbres de preuves) sur le Web, à l'aide de représentations de ces preuves en XML et MathML otenues à partir de notre interface Pcoq. Ces représentations et les formules mathématiques associées sont destinées à être visualisées par les navigateurs usuels, les rendant ainsi indépendantes de Pcoq. Ce travail s'effectue actuellement dans le cadre du contrat européen LTR Mowgli.

A long terme, l'évolution vers les standards issus du Web devrait augmenter la visibilité de notre travail, en augmentant le nombre d'utilisateurs de nos outils.

5. Logiciels

5.1. Pcoq

Participant : Ahmed Amerkad, Yves Bertot [correspondant], Loïc Pottier, Laurence Rideau.

La version 1.3 de pcoq a été rendue disponible en janvier 2002. L'interface est plus robuste et utilisable avec une plus large palette de versions du langage Java.

Une nouvelle mouture du logiciel est en cours d'élaboration (elle a été utilisée dans nos expériences sur le théorème de Bertrand).

5.2. Aïoli et Figue

Participant : Hanane Naciri, Laurence Rideau, Laurent Théry [correspondant].

Aïoli (http://www-sop.inria.fr/croap/aioli/doc/aioli.html) et Figue (http://www-sop.inria.fr/croap/figue/) sont des composants de base de Pcoq (Aïoli pour le traitement des arbres, et Figue pour leur affichage), et à ce titre ont reçu des améliorations : sélections multiples, nouveaux combinateurs 2D (matrices, crochets, fractions, racines n-ièmes).

6. Résultats nouveaux

6.1. Outils pour les environnements de preuve

6.1.1. Evolution du système Pcoq

Participants : Ahmed Amerkad, Janet Bertot, Yves Bertot, Hanane Naciri, Laurence Rideau, Loïc Pottier, Laurent Théry.

Une nouvelle version de l'interface graphique pour le système de preuve Coq a été distribuée en cours d'année. Des travaux ont été menés pour améliorer l'interface sur plusieurs points : - L'intégration de la communication par les formats standards de style XML, MathML a été améliorée.

- Une expérience a été menée dans le cadre du stage de fin d'études de Florence Martel (élève ingénieur du CNAM) pour intégrer des outils d'analyse de structure des scripts de preuve au niveau des commandes élémentaires de preuve, en application des études effectuées dans le passé au cours de la thèse d'Olivier Pons.

- Une expérience a été menée dans le cadre du stage de première année de Sophie Boulmé (élève ingénieur de l'ESINSA) pour construire un outil de ré-exécution de scénarios d'utilisation pour l'interface graphique. Nous envisageons que cette expérience pourra mener à la constitution d'un outil important pour la stratégie d'assurance qualité du logiciel, en permettant de ré-exécuter les scénarios où apparaissent des problèmes.

- Janet Bertot a travaillé à la vérification de qualité pour toutes les procédures d'annulation de commandes de l'outil interactif. Les résultats de ce travail menent à une interface beaucoup plus robuste.

6.1.2. Interaction homme machine dans les environnements de preuves

Participants : Laurence Rideau, Hanane Naciri.

Ce travail se place dans le cadre du développement d'outils pour l'interaction homme machine dans les environnements de démonstration mathématique. Nous continuons à étendre et à améliorer notre outil d'affichage bidimensionnel, incrémental et interactif FIGUE. Cet outil, permet de manipuler dynamiquement les objets structurés représentant des documents comme des programmes ou des formules mathématiques. Notre but est de présenter les objets structurés et en particulier les formules mathématiques de façon conviviale et d'offrir des interactions variées à la souris comme la sélection de sous-expressions afin de pouvoir les manipuler dynamiquement (évaluation, simplification, modification, génération de code, ...). Nous étudions le problème lié à la diversité des objets à manipuler : du texte simple, des formules mathématiques pour les systèmes de preuves ou de calcul formel, des images. Des problèmes particuliers se posent pour les formules mathématiques qui ont une structure complexe et bidimensionnelle (matrices, intégrales, indices, ...).

Cette année, nous avons travaillé plus particulièrement sur l'affichage bidirectionnel des formules mathématiques dans différents systèmes d'écriture comme l'arabe et l'hébreu. La manipulation des formules mathématiques dans un contexte bidirectionnel a un niveau de complexité plus élevé que la manipulation du texte simple (linéaire). Cela est dû à la nature bidimensionnelle de formules mathématiques et à l'importance des relations spatiales dans les notations mathématiques. La profondeur d'imbrication des objets dans une formule augmente la difficulté de mélanger les deux directions droite à gauche et gauche à droite au sein de la même formule. Dans notre étude, nous distinguons deux cas de formules mathématiques :

- les formules écrites suivant le sens de déroulement de l'écriture du texte qui les contient (par exemple, les formules à l'égyptienne écrites de droite à gauche incluses dans le texte arabe)
- les formules écrites dans le sens inverse du texte qui les contient (par exemple, les formules à la marocaine écrites de gauche à droite à l'inverse du texte arabe).

Notre but est de proposer une approche générale pour manipuler les formules mathématiques dans un contexte bidirectionnel (mélange des directions d'écritures) et de déterminer les règles de direction spécifiques aux objets mathématiques. Nous nous sommes basés sur cette technique pour étendre notre moteur d'affichage FIGUE et offrir un affichage bidirectionnel du document : d'abord un affichage de droite à gauche (par exemple l'affichage du texte arabe) et ensuite un affichage bidirectionnel mélangeant à la fois la direction de gauche à droite et la direction de droite à gauche (par exemple du texte arabe incluant du texte indo-européen). Le résultat de ce travail est utilisé pour présenter des explications de preuves mathématiques en arabe dans le système PCOQ.

Ce travail représente aussi une base d'expérience pour définir l'usage de la norme MathML dans le contexte de l'affichage bidirectionnel de formules mathématiques. Le support de cette utilisation de MathML par des navigateurs Web permettrait une large diffusion de documents scientifiques sur le Web en différentes langues.

6.1.3. Tracé d'énoncés géométriques

Participants : Frédérique Guilhot, Farid Latrech, Loïc Pottier.

Farid Latrech, durant son stage en juillet, a programmé une interface permettant de tracer des énoncés géométriques écrits dans la syntaxe de Coq. Ce tracé se fait grâce à une applette Java, GeoPlanJ, qui est une implémentation en Java (encore partielle) du logiciel GeoPlanW (développé par le CREEM du CNAM, et utilisé dans l'enseignement secondaire). Le travail de Farid Latrech a été repris et intégré dans Pcoq : on peut désormais tracer une figure correspondant à un énoncé géométrique, ou bien correspondant aux hypothèses courante lorsqu'on développe une preuve.

6.1.4. Visualisation de formules MathML avec Figue et le Web

Participants : David Baccialon, Loïc Pottier.

Durant son stage d'été, David Baccialon a expérimenté plusieurs méthodes pour visualiser des formules mathématiques en MathML grâce à Figue. La solution utilisant des applettes s'est avérée impraticable car trop lourde (les navigateurs comme IE6, netscape, Mozilla, Amaya sont incapables de traiter correctement plus de deux applettes visualisant chacune une formule). La solution qui fonctionne consiste à générer au vol (par un serveur Web), à partir d'une page html contenant du MathML, une nouvelle page html où les formules MathML sont transformées en images associées à des « map » cliquables : en cliquant sur une sous-formule, on obtient dans le presse-papier du système son code MathML.

6.1.5. Exportation de preuves Coq en MathML

Participants : Loïc Pottier.

Dans le cadre des tâches qui nous sont dévolues dans le projet Mowgli, nous avons développé un programme dans Coq permettant de traduire un arbre de preuve (en format interne à Coq) en XML. Le but de ce travail est de pouvoir ensuite traiter cet arbre XML pour le visualiser soit sous forme de script Coq, soit en langue naturelle.

6.1.6. Environnement de preuves intégré à TeXmacs

Participants : Philippe Audebaud, Laurence Rideau.

GNU TeXmacs est un éditeur de texte scientifique libre. Il permet d'écrire des documents structurés en mode dit *wysiwyg*. Le programme utilise à l'affichage les mêmes polices et les mêmes algorithmes de mise en page que TEX. Il peut être facilement étendu par l'utilisateur via l'écrire de nouveaux styles ou de scripts écrits dans le langage Scheme.

Ainsi, par beaucoup de ses caractéristiques, TeXmacs se prête à son expérimentation comme environnement de travail pour le système de preuve Coq. Nous nous proposons de suivre la progression suivante :

Documentation : en étendant les mécanismes existants d'importation et d'exportation avec d'autres formats, permettre la documentation des scripts de preuves Coq et l'édition d'un article au format LATEX à partir des données contenues dans ces fichiers scripts.
Ce mode d'utilisation est actuellement réalisé ; il s'appuie sur la même représentation du script Coq que celle utilisée par Pcoq pour structurer le contenu. La présentation dans l'éditeur est prise en compte par des fichiers de style entièrement configurables par l'utilisateur.

Session de travail Coq : utiliser le mode Session de TeXmacs pour communiquer avec le toplevel Coq et développer des preuves de manière interactive dans l'environnement d'un éditeur ayant les fonctionnalités de GNU Emacs.
Ce mode de travail est déjà possible. Ses imperfections tiennent à la dispersion des appels aux commandes d'affichage dans le code Coq qui nécessiterait de restructurer un grand nombre de fichiers dans le source Coq. A cette occasion, l'effort est mis sur la modularité du code de Coq pour regrouper les fonctions d'entrée-sortie.

Environnement de développement intégré : dans lequel le processus Coq est rendu totalement invisible à l'utilisateur, au profit d'un mode d'édition plus proche de la feuille de papier.

La dernière partie nécessite encore de comprendre dans quelle direction il convient de porter les efforts. L'environnement Pcoq offre nombre de fonctionnalités qui vont dans la même direction. Ces deux environnements apparaissent complémentaires ; un environnement de développement intégré pour les preuves mathématiques basé sur Coq ne peut que bénéficier de l'expérience acquise dans ces deux outils logiciels.

6.2. Théorie des types et formalisation de théories mathématiques

6.2.1. Fonctions récursives

Participants : Antonia Balaa, Gilles Barthe, Yves Bertot, Venanzio Capretta, Pierre Courtieu, Kuntal Das Barman, Nicolas Magaud.

Les travaux d'Antonia Balaa et Yves Bertot sur la description de fonctions récursives dans le calcul des constructions à partir de leur équation de point-fixe ont donné lieu cette année à la publication d'un article [10] et à la soutenance de thèse d'Antonia Balaa. Un prototype décrivant ces techniques a été développé et expérimenté sur un petit nombre d'exemples. Des extensions ont été proposées par Yves Bertot, Kuntal Das Barman, et Venanzio Capretta pour permettre également la manipulation dans le calcul des constructions de fonctions récursives partielles présentant une récursion imbriquée. Un article sur ce sujet a été publié [18].

Les travaux de Nicolas Magaud sur la traduction de démonstrations d'une structure à une autre ont progressé grâce à une meilleure compréhension des techniques de démonstrations sur les fonctions récursives bien fondées, car les fonctions structurelles récursives pour une structure de donnée correspondent généralement à des fonctions récursives bien fondées pour une autre structure de donnée. L'évolution récente se concentre sur une meilleure compréhension des preuves d'égalité.

Yves Bertot a participé avec Pierre Castéran de l'université de Bordeaux à la rédaction d'un livre sur l'utilisation du système de démonstration Coq dans la production de logiciel certifié.

6.2.2. Théorie des Types

Participant : Gilles Barthe.

En collaboration avec H. Cirstea, C. Kirchner et L. Liquori, nous avons défini une extension des systèmes de types purs (PTSs) avec des motifs. Ces travaux ont donné lieu à une publication dans une conférence internationale.

6.2.3. Raisonnement inductif dans le μ-calcul

Participants : Christoph Sprenger, Mads Dam (KTH, Suède).

Nous avons étudié les mécanismes du raisonnement inductif dans le cadre du μ-calcul du premier ordre. En particulier, nous avons comparé deux systèmes de preuve. Le premier utilise une règle traditionelle d'induction bien-fondée et produit des preuves en forme d'arbre (induction locale), tandis que le deuxième construit des preuves en forme de graphes et emploie une condition de décharge externe pour justifier la bonne-fondation des arguments inductifs (induction globale). Notre étude a démontré l'équivalence des deux types d'induction. Un papier décrivant nos résultats a été soumis à la conférence FOSSACS/ETAPS 2003. Un résumé de travail situé dans ce même contexte a été presenté au workshop « Fixed Points in Computer Science 2002 » en mois de juillet 2002 à Copenhague. Une version étendue a été soumis à la revue Informatique Théorique et Applications.

Nous proposons de continuer ces recherches dans le cadre de la théorie des types et plus particulèrement dans le calcul des constructions inductives. Parallèlement, on envisage l'étude de la vérification compositionnelle de code mobile et embarqué en collaboration avec le groupe FDT au SICS.

6.2.4. Géométrie du lycée.

Participant : Frédérique Guilhot, Loïc Pottier.

Dans un premier temps, nous avons formalisé en Coq la théorie élémentaire des angles orientés de vecteurs non nuls du plan euclidien. Ce qui a permis démontrer des théorèmes classiques : le théorème qui donne une condition nécessaire et suffisante pour que quatre points soient cocycliques, le théorème qui montre que les symétriques de l'orthocentre d'un triangle par rapport à ses côtés sont sur son cercle circonscrit, le théorème de la droite de Simson et le théorème de Napoléon. Ce développement a été fait en utilisant l'interface Pcoq, avec des notations mathématiques classiques (voir le rapport de recherche [31]).

Dans un deuxième temps, le développement en Coq d'un cours avec exercices de géométrie affine euclidienne à l'usage des lycéens a été commencé. Les « chapitres » suivants ont été abordés mais sont encore en cours de développement :

- notions de points, de vecteurs ;
- barycentre : définition et propriétés ;
- alignement, droites parallèles et concourantes ;
- théorèmes de Thalès, de Desargues, du trapèze complet ;
- étude des translations et des homothéties ;
- plans et droites dans l'espace : incidence et parallélisme ;
- produit scalaire et orthogonalité ;
- orthogonalité dans l'espace ;
- application : résolution d'un exercice donné au baccalauréat S ;
- distance euclidienne ;
- médiatrice, triangles isocèles, cercles ;
- angles de vecteurs, angles de droites dans le plan ;
- trigonométrie ;
- étude des rotations planes
- étude des réflexions planes
- étude des composées des transformations ;
- théorème de cocyclicité et applications (voir la partie concernant les angles ci-dessus).

6.2.5. Quotients

Participant : Laurent Chicli, Loïc Pottier, Carlos Simpson.

Grâce à un contre-exemple de Carlos Simson, nous nous sommes aperçus que l'axiome que nous avions proposé pour traiter correctement les quotients d'espaces de fonctions était contradictoire en Coq (à cause de la sorte Set et de son imprédicativité). Ce contre-exemple a été simplifié par Laurent Chicli, et nous avons ensuite étudié en détail les divers paradoxes et inconsistances qu'il permettait de simplifier ou de démontrer. La conclusion nous semble être qu'en l'état, la théorie de Coq ne permet pas de traiter sans problème de consistance les mathématiques classiques.

Ce travail a été exposé dans un workshop et est soumis à publication [21].

6.2.6. Sémantique axiomatique des algorithmes probabilistes

Participant : Philippe Audebaud.

Nous avons proposé un système d'inférence en sémantique axiomatique propre à spécifier et développer des preuves portant sur des programmes impératifs (tels que proposés par C.A.R Hoare) comportant en plus des instructions appelant des générateurs aléatoires.

Le travail de formalisation dans Coq sous la forme d'une contribution utilisateur requiert des développements récents de l'assistant de preuve : disponibilité d'une riche librairie de résultats en analyse réelle et élargissement de la notion d'anneau à des structures dans lesquelles l'égalité est définissable (par opposition à l'égalité de Liebniz). Nous avons aussi pris le parti de raisonner sur des objets construits comme des paires constituées d'un témoin calculatoire et d'une preuve que ce témoin satisfait telle propriété demandée. Cette structure de sous-ensemble s'appuie sur une utilisation contrôlée des coercions de Coq ; elle se prête particulièrement bien au contexte sémantique, où nous devons raisonner sur des fonctions, des fonctionnelles, etc. En faisant hériter la structure d'ω-cpo dont découle la preuve d'existence des diverses sémantiques introduites. L'architecture de ce développement est stable ; l'ensemble des scripts est à ce jour incomplet.

6.3. Certification d'algorithmes

6.3.1. Racine carrée

Participant : Yves Bertot, Nicolas Magaud, Mélanie Vaillant, Paul Zimmerman (projet Spaces).

Les travaux de Nicolas Magaud, Yves Bertot et Paul Zimmermann sur la certification d'algorithmes de racines carrées ont donné lieu à la publication d'un article dans un journal international [9]. Ces travaux ont été prolongés par la formalisation d'un algorithme simple de calcul de racine cubique et d'un algorithme général de calcul de racine énième, dans le cadre du stage d'été de Mélanie Vaillant. Une implémentation efficace et certifiée pour la racine cubique est également à l'étude.

6.3.2. Arithmétique de Presburger

Participant : Laurent Théry.

Nous nous sommes interessé à la formalisation dans le système Coq du papier fondateur de Presburger (1932). Utilisant le mécanisme de réflexion de Coq, cette formalisation nous fournit directement une procédure de décision (tactique) pour l'arithmétique entière sans multiplication. Notre formalisation a été intégrée aux contributions distribuées avec le système Coq.

6.3.3. Calcul de nombres premiers et théorème de Bertrand

Participant : Laurent Théry.

Le théorème de Bertrand assure qu'il existe toujours au moins un nombre premier entre un nombre (plus grand que 2) et son double. La preuve de ce théorème, proposée par Erdös, a été entièrement formalisée dans le système Coq. Ce théorème est la propriété clé qui permet de vérifier formellement l'algorithme proposé par Knuth pour calculer les n premiers nombres premiers. Ce travail a fait l'objet d'un rapport de recherche ainsi que d'une contribution au système Coq [32].

6.3.4. Elimination des quantificateurs

Participants : Assia Mahboubi, Loïc Pottier.

Le travail commencé en 2001 sur la mise au point d'une tactique en Coq permettant de faire des preuves sur les inégalités polynomiales à coefficients réels s'est poursuivi cette année. Une communication a été présentée à JFLA2002 [26]. Le cas à une variable est complètement programmé, et fonctionne en Coq sur un prototype. Le cas général est compris en théorie, et reste à programmer, pour que la tactique puisse être fournie ensuite dans la distribution de Coq, complétant ainsi la tactique Fourier qui traite déjà les cas linéaires.

D'un point de vue plus théorique, on cherche à améliorer l'algorithme en utilisant des sous-résultants dans les divisions.

6.4. Sémantique des langages de programmation

6.4.1. Description sémantique des langages

Participant : Yves Bertot.

Les techniques de description de fonctions récursives générales ont été appliquées à la description fonctionnelle de la sémantique d'un petit langage. En particulier, des efforts ont été réalisés pour contourner la limitation du calcul des constructions qui impose que l'on ne manipule que des fonctions totales. Le résultat est que l'on est capable de construire un « interprète fonctionnel certifié » par rapport à une description de sémantique naturelle (ce travail a déjà été mentionné dans la section sur les fonctions récursives).

6.4.2. Le langage JavaCard

Participants : Gilles Barthe, Pierre Coutieu, Guillaume Dufay, Simão Melo de Sousa.

En collaboration avec l'Université de la Républica, Montevideo, nous avons formalisé un algorithme d'élimination des sous-routines, et établi sa correction par rapport à la sémantique opérationnelle de la machine virtuelle JavaCard. Nous nous sommes également intéressés au mécanisme de firewall de JavaCard, et avons proposé une méthodologie pour remédier à ses faiblesses.

Nous avons également poursuivi nos travaux sur la modélisation de la plateforme JavaCard, et nous avons en particulier développé un cadre générique pour valider des vérificateurs de bytecode dans Coq.

6.4.3. Environnement de vérification pour JavaCard

Participants : Gilles Barthe, Pierre Coutieu, Guillaume Dufay, Simão Melo de Sousa.

Nous avons poursuivi nos travaux sur Jakarta, en nous attachant plus particulièrement à l'automatisation des preuves. Le travail a été effectué dans le contexte de Spike et de Coq. Les résultats obtenus sont encourageants : les preuves de validation croisées des machines virtuelles peuvent être automatisées dans une large mesure.

6.4.4. Vérification du code mobile et embarqué

Participant : Gilles Barthe.

Dans le cadre de notre participation au projet Profundis, nous avons étudié les systèmes de types pour la confidentialité des données dans un langage concurrent. En particulier, nous avons formalisé un système de types proposé par G. Boudol et I. Castellani, et établi sa correction en Coq.

6.4.5. Langages de spécification

Participants : Néstor Cataño Collazos, Marieke Huisman, Kerry Trentelman.

Nous avons continué l'étude des langages de spécification pour Java, comme JML et ESC/Java. Notre recherche peut être divisée en deux parties :

1. l'étude des langages existants (utilité, techniques pour la vérification, etc.) ;
2. proposition d'une extension des langages existants, pour permettre d'exprimer plus des propriétés.

Ad. 1, nous avons pris un cas d'étude industriel (de Gemplus), une implémentation d'un porte-monnaie électronique, et nous avons écrit et validé des spécifications fonctionnelles pour cette implémentation, en utilisant l'outil ESC/Java (développé à Compaq Research). ESC/Java est un outil qui essaie de trouver automatiquement les erreurs les plus communes dans les programmes Java. Nous avons montré qu'on peut aussi utiliser l'outil pour valider des spécifications fonctionnelles. Dans le cas d'étude nous avons trouvé un grand nombre d'erreurs.

Pendant ce travail, nous avons réalisé qu'une omission importante de ESC/Java était que l'outil ne validait pas la clause *assignable*, qui spécifie quelles variables peuvent être modifiées par une méthode. Pour cette raison, nous avons développé notre propre outil, baptisé « Chase », qui fait cette validation.

Ad. 2, nous avons proposé une extension de langage JML avec constructions temporelles. Dans cette extension de JML, on peut exprimer par exemple qu'on doit appeler certaines méthodes dans une ordre particulier, ou qu'après l'appel d'une méthode une certaine condition est devenue un invariant de la classe.

6.4.6. Vérification des programmes concurrents

Participants : Néstor Cataño Collazos, Marieke Huisman.

Dans le cadre de la thèse de Néstor Cataño, nous avons étudié le problème de la vérification de programmes concurrents (dont Java et JavaCard) de manière efficace. En particulier, nous travaillons au développement d'une plate-forme de vérification où l'on pourrait faire converger les techniques de *theorem proving* et *model-checking*. Pour cela, les *espaces d'événements*, tels qu'ils ont été introduits par Cenciarelli *et al.* pour modéliser la mémoire de Java, ont étés envisagés comme un modèle valide pour interpréter et vérifier des propriétés de programmes concurrents écrits en Java. Actuellement nous sommes capables de générer un espace d'événements à partir d'un programme Java.

6.4.7. Vérification compositionelle du code mobile et embarqué

Participants : Gilles Barthe, Gennady Chuganov (SICS,Suède), Lars-Åke Fredlund (SICS, Suède), Dilian Gurov (SICS, puis KTH, Suède), Marieke Huisman.

Nous avons étendu notre travail sur la vérification de comportement des programmes. Nous avons amélioré notre cadre de travail qui consiste en un modèle abstrait des programmes Java, un langage de spécification et une logique pour la vérification. Maintenant, le modèle de programme est plus proche du programme original. Puis nous avons construit des outils pour la validation d'une applette isolée. Finalement, nous avons travaillé sur le problème de décidabilité de la correction d'une décomposition [15][30].

7. Contrats industriels

7.1. Dassault Aviation

Ce contrat finance la thèse d'Antonia Balaa, sur l'étude des fonctions récursives à terminaison non structurelle en théorie des types.

7.2. Mowgli

La proposition européenne Mowgli (LTR) a démarré cette année. Le sujet concerne les mathématiques formelles sur le Web. Participants : Universités de Bologne, de Berlin, de Nijmegen et Eindhoven, DFKI Saarbrücken, Max Planck Institute et la société Trusted Logic.

7.3. Verificard

Ce contrat, démarré en 2001 pour une durée de 3 ans, finance les thèses de Kuntal das Barman, de Nestor Cataño Collazos et de Kwong Cheong Wong, et les stages post-doctoraux de Sorin Stratulat et Pierre Courtieu. Il concerne la modélisation et la vérification de la plate-forme et des programmes Javacard (participants :

Schlumberger, Universités de Nijmegen, Munich, Kaiserslautern, Sics). Le travail du projet dans ce cadre est décrit dans la partie « Sémantique des langages de programmation ».

8. Actions régionales, nationales et internationales

8.1. Collaborations internationales

- Ontario Research Center for Computer Algebra, Canada : sur MathML (affichage bi-directionnel de formules mathématiques).
- Kent State University, United States : calcul mathématique via internet (affichage interactif de formules).
- Université du Minho, Portugal, (INRIA-ICCTI) et Technical University of Tallinn, Estonia : terminaison des fonctions récursives basée sur les types, constructor subtyping, CPS translations of inductive types.
- Université de Nijmegen, the Netherlands : formalisation des mathématiques en théorie des types (certification de la transformée de Fourier rapide, sétoïdes en théorie des types).
- Chalmers University of Technology : récursion générale en théorie des types.
- Université de Córdoba, Argentina, Université de Montevideo, Uruguay (MAE funded) : méthodes formelles pour les cartes à puces (vérification de la plate-forme JavaCard en Coq, non-interférence d'applettes).
- SICS et KTH, Stockholm, Sweden : vérification compositionnelle de systèmes pour la vérification de propriétés de sécurité.
- Université de Nijmegen, the Netherlands, Université de Kaiserslautern, Germany, Australian National University, Canberra, Australia : langages de spécification pour java (extension de JML avec logique temporelle).
- Université de Bologne, de Nijmegen, German Research Center for Artificial Intelligence DFKI et Max Planck Institute for Gravitational Physics : mathématiques sur le Web.

8.2. Actions nationales

8.2.1. Action de Recherche Coopérative ModoCop

L'année 2002 est la première année de l'action de recherche coopérative ModoCop (*ModelChecking Object-Oriented Programs*). Cette action regroupe les équipes Lande (Rennes), Lemme (Sophia), Oasis (Sophia), Vasy (Grenoble), Vertecs (Rennes) et le groupe *Distributed and Complex Systems Research* de Verimag. Elle a pour but la spécification, la vérification automatique et le test symbolique des programmes orientés objet concurrents. La première réunion s'est tenue en mars à Sophia, la deuxième en juillet à Rennes et la troisième en novembre à Grenoble. L'activité de cette action de recherche est présentée à l'adresse suivante : (http://www-sop.inria.fr/lemme/modocop/.

8.3. Actions européennes

8.3.1. Réseau Types

Lemme participe activement au réseau Types reconduit en 2000, qui regroupe les équipes européennes travaillant sur la théorie des types.

9. Diffusion des résultats

9.1. Manifestations scientifiques, missions

Antonia Balaa, Yves Bertot, Nicolas Magaud, Assia Mahboubi, Laurence Rideau, ont participé à la conférence JFLA2002 à Anglet les 28-29 janvier : Antonia Balaa y a présenté ses travaux sur les

fonctions récursives et Yves Bertot y a présenté ses travaux sur la description formelle de logiciel lors d'un exposé invité. Assia Mahboubi a présenté ses travaux avec Loïc Pottier sur une tactique Coq d'élimination des quantificateurs sur les nombres réels.

Philippe Audebaud, Yves Bertot, Venanzio Capretta, Laurent Chicli ont participé au colloque Types à Nijmegen (Pays-Bas) du 24 au 28 avril 2002.

Gilles Barthe, Yves Bertot et Venanzio Capretta ont participé au Colloque Types sur la termination à Göteborg (Suède) les 14 et 15 novembre 2002. Venanzio Capretta y a présenté ses travaux sur la représentation de fonctions récursives à l'aide de types co-inductifs. Yves Bertot y a présenté les travaux qu'il avait effectués sur les fonctions récursives imbriquées avec Venanzio Capretta et Kuntal Das Barman.

Gilles Barthe, Yves Bertot, Venanzio Capretta, Pierre Courtieu,
Kuntal Das Barman et Laurent Théry ont participé à la conférence internationale TPHOLS'02 qui a eu lieu à Hampton, Virginia (États-Unis) du 19 au 23 août 2002. Pierre Courtieu y a présenté les travaux qu'il avait effectués avec Gilles Barthe. Kuntal Das Barman y a présenté les travaux qu'il avait effectués avec Yves Bertot et Venanzio Capretta.

Yves Bertot a rendu visite à la Nasa-Langley à Hampton, Virginie (États-Unis) pour y présenter les travaux de l'équipe Lemme sur la formalisation du langage JavaCard.

Yves Bertot a participé avec Joëlle Despeyroux (équipe Miró) et Jean Duprat (école normale supérieure de Lyon) à l'organisation de l'école d'été pour jeunes chercheurs organisée à Giens du 2 au 13 septembre 2002.

Yves Bertot et Loïc Pottier ont participé à une réunion Mowgli (Mathematics on the Web) à Eindhoven les 18-19 juillet 2002.

Néstor Cataño a présenté un exposé à la conférence FME (Copenhague, Danemark), et pendant une réunion ModoCop. Il a assisté aux écoles d'été à Nantes et Marktoberdorf, à la réunion de VerifiCard au Bandera et au Spin Workshop (Grenoble), pendant Etaps.

Marieke Huisman a présenté un exposé aux conférences FASE (Grenoble), MPC (Dagstuhl, Allemagne), AMAST (Réunion). Elle a assisté aux FME et CAV (Copenhague, Danemark). Elle a donné des exposés pendant les réunions ModoCop et VerifiCard et pendant une réunion de projet Mobi-J (collaboration Allemagne-Pays Bas).

Hanane Naciri a participé au 6ème Colloque Africain sur la Recherche en Informatique à Yaoundé, Cameroun, du 14 au 17 octobre.

Hanane Naciri et Laurence Rideau ont participé à la Conference Internationale MathML(MathML'2002) à Chicago, USA, du 28 au 30 juin.

Guillaume Dufay a donné un exposé au Java Workshop Verification à Portland (États-Unis) et à la conférence POPL du 14 au 18 janvier.

Laurent Théry a donné deux exposés invités (Formalisation des mathématiques et Comparaison des systèmes de preuve HOL et PVS) à l'école d'été pour jeunes chercheurs organisée à Giens.

9.2. Animation de la communauté scientifique

- Laurence Rideau a présidé le comité de programme et l'organisation de la conférence JFLA2002 à Anglet les 28 et 29 janvier.
- Laurent Théry a co-organisé avec Andrew Adams (University of Reading), Hanne Gottliebsen (ICASE) et John Harrison (Intel Corporation) un colloque sur la formalisation de mathématiques continues à Hampton, Virginia (États-Unis).
- Marieke Huisman a mis au point le programme pour la réunion VeriSafe à Nice (réunion commune du projet européen SecSafe et VerifiCard) en collaboration avec Thomas Jensen (Lande, Rennes). Ils sont co-éditeurs d'un numéro spéciale du *Journal of Logic and Algebraic Programming* autour des méthodes formelles pour la carte à puce.
- Yves Bertot était membre du comité de programme de JFLA2002.

- Les membres du projet ont évalué des articles pour JAR, JFP, TCS, JFLA Types, ECAI, JACO, ICALP.

9.3. Divers

Yves Bertot est membre de la commission de spécialistes 27e section à l'université de Marseille II et membre suppléant de la commission de spécialistes 27e section à l'école normale supérieure de Lyon.

Loïc Pottier est membre nommé du CNU 27ème section, membre suppléant de la commission de spécialistes 27ème section l'université de Perpignan.

9.4. Visites

Nous avons accueilli de nombreux visiteurs étrangers, en particulier à l'occasion du séminaire du projet (cf http://www-sop.inria.fr/lemme/Nicolas.Magaud/seminaire/index.html)

Daniel Perovich (INCO, Uruguay), jusqu'à mars 2002.
Daniel Fridlender (Cordobà, Argentine), mars - avril 2002.
Leonardo Rodrìguez (INCO, Uruguay), février - mai 2002.
Dilian Gurov (KTH, Suède), mars 2002, une semaine.
Claudio Sacerdoti-Coen (Bologne).

9.5. Direction de thèses

Gilles Barthe dirige les thèses de Maria João Frade (Université du Minho, Portugal), Simão Melo de Sousa, Guillaume Dufay et Kwong Cheong Wong.

Yves Bertot dirige les thèses d'Antonia Balaa, Nicolas Magaud et Kuntal Das Barman.

Marieke Huisman dirige la thèse de Nestor Cataño Collazos.

Loïc Pottier codirige avec André Hirschowitz la thèse de Laurent Chicli.

Laurence Rideau dirige la thèse de Hanane Naciri.

9.6. Jury de thèses

Yves Bertot a été membre du jury de la thèse de Ludovic Casset (GemPlus et université de Marseille II) le 2 octobre 2002 et du jury de thèse d'Antonia Balaa le 6 novembre 2002.

Yves Bertot a participé à la « defense » de la thèse d'Ana Bove (Chalmers University, Göteborg, Suède) en tant qu'« opponent », le 8 novembre 2002.

Yves Bertot a été rapporteur de la thèse de Cuihtlauac Alvarado (France-Télécom et université de Paris-Sud), le 18 décembre 2002.

9.7. Encadrement de stagiaires

Yves Bertot : Florence Martel CNAM, 6 mois, Sophie Boulmé ESINSA, 2 mois, Mélanie Vaillant Maîtrise de mathématiques-informatiques 6 semaines, Franck Courtès et Jean-Charles Leroy ESSI, 2 mois.

Loïc Pottier : David Baccialon, 4 mois, Farid Latrech 3 mois, Radia Abib 2 mois (tous trois de la MIM de Nice).

Laurence Rideau : Sébastien Marti IIE (Evry), 6 mois co-encadré avec CAFÉ

9.8. Enseignement

Gilles Barthe : cours de DEA Info. à l'UNSA (25h).
Yves Bertot : Sémantique des langages de programmation en Maîtrise (18 heures), Introduction aux méthodes formelles en DESS (24 heures), Programmation fonctionnelle et preuves en 1ère année de Magistère (32 heures).
Guillaume Dufay : 12 heures de TD de Sémantique des langages de programmation en Maîtrise, 39 heures de TD d'Informatique théorique en 2è année de DEUG MASS.
Marieke Huisman : TD de Sémantique des langages de programmation en Maîtrise.
Loïc Pottier : 52h de cours d'algorithmique et logique en Maîtrise MIM à l'UNSA.
Laurent Théry : 48h de cours de langages de programmation avancés, 24h de cours de méthodes formelles à l'université de L'Aquila (Italie).

10. Bibliographie

Livres et monographies

[1] *Applied Semantics, International Summer School, APPSEM 2000, Caminha, Portugal, September 9-15, 2000, Advanced Lectures.* éditeurs G. BARTHE, P. DYBJER, L. PINTO, J. SARAIVA., série Lecture Notes in Computer Science, volume 2395, Springer, 2002, http://dblp.uni-trier.de.

Thèses et habilitations à diriger des recherche

[2] A. BALAA. *Fonctions récursives générales dans le calcul des constructions.* thèse de doctorat, Université de Nice-Sophia Antipolis, 2002.

[3] V. CAPRETTA. *Abstraction and Computation.* thèse de doctorat, Computing Science Institute, University of Nijmegen, 2002.

[4] H. NACIRI. *Conception et Réalisation d'outils pour l'interface homme-machine des environnements pour les mathématiques.* thèse de doctorat, Université de Nice-Sophia Antipolis, 2002.

Articles et chapitres de livre

[5] A. ARMANDO, M. RUSINOWITCH, S. STRATULAT. *Incorporating Decision Procedures in Implicit Induction.* in « Journal of Symbolic Computation », numéro 4, volume 34, 2002, pages 241-258.

[6] G. BARTHE, V. CAPRETTA, O. PONS. *Setoids in Type Theory.* in « Journal of Functional Programming », 2002.

[7] G. BARTHE, T. COQUAND. *On the equational theory of non-normalizing Pure Type Systems.* in « Journal of Functional Programming », 2002, To appear.

[8] G. BARTHE, M. J. FRADE, E. GIMÉNEZ, L. PINTO, T. UUSTALU. *Type-Based Termination of Recursive Definitions.* in « Mathematical structures in Computer Science », 2002, To appear.

[9] Y. BERTOT, N. MAGAUD, P. ZIMMERMANN. *A GMP program computing square roots and its proof within Coq.* in « Journal of Automated Reasoning », 2002, To appear. Special Issue on Automating and Mechanising Mathematics : In honour of N.G. de Bruijn.

Communications à des congrès, colloques, etc.

[10] A. BALAA, Y. BERTOT. *Fonctions récursives générales par itération en théorie des types.* in « *Journées Francophones pour les Langages Applicatifs* », janvier, 2002.

[11] G. BARTHE, T. COQUAND. *An Introduction to Dependent Type Theory.* in « Applied Semantics. Lecture Notes for the APPSEM Summer School », série Lecture Notes in Computer Science, volume 2395, Springer-Verlag, éditeurs G. BARTHE, P. DYBJER, L. PINTO, J. SARAIVA., 2002.

[12] G. BARTHE, P. COURTIEU. *Efficient Reasoning about Executable Specifications in Coq.* in « Proceedings of TPHOLs'02 », série Lecture Notes in Computer Science, Springer-Verlag, éditeurs C. M. V. CARREÑO, S. TAHAR., 2002.

[13] G. BARTHE, P. COURTIEU, G. DUFAY, S. M. DE SOUSA. *Tool-Assisted Specification and Verification of the JavaCard Platform.* in « Algebraic Methodology And Software Technology (AMAST '02) », série Lecture Notes in Computer Science, volume 2422, Springer-Verlag, éditeurs H. KIRCHNER, C. RINGESSEIN., 2002.

[14] G. BARTHE, G. DUFAY, L. JAKUBIEC, S. M. DE SOUSA. *A formal correspondence between offensive and defensive JavaCard virtual machines.* in « Proceedings of VMCAI'02 », série Lecture Notes in Computer Science, volume 2294, Springer-Verlag, éditeurs A. CORTESI., pages 32-45, 2002.

[15] G. BARTHE, D. GUROV, M. HUISMAN. *Compositional Verification of Secure Applet Interactions.* in « Proc. FASE'02 », série Lecture Notes in Computer Science, numéro 2306, Springer-Verlag, éditeurs R.-D. KUTSCHE, H. WEBER., pages 15-32, 2002.

[16] G. BARTHE, T. UUSTALU. *CPS Translating Inductive and Coinductive Types.* in « Proceedings of PEPM'02 », ACM Press, éditeurs P. THIEMANN., 2002.

[17] Y. BERTOT. *Des descriptions fonctionnelles aux implémentations impératives de programmes.* in « *Journées francophones pour les langages applicatifs, JFLA'02* », janvier, 2002, En Français.

[18] Y. BERTOT, V. CAPRETTA, K. D. BARMAN. *Type-theoretic functional semantics.* in « Theorem Proving in Higher Order Logics (TPHOLS'02) », série LNCS, numéro 2410, Springer-Verlag, août, 2002.

[19] N. CATAÑO, M. HUISMAN. *Chase : a Static Checker for JML's Assignable Clause.* in « Verification, Model Checking and Abstract Interpretation (VMCAI '03) », LNCS. Springer, 2002.

[20] N. CATAÑO, M. HUISMAN. *Formal Specification and Static Checking of Gemplus's Electronic Purse Using ESC/Java.* in « Formal Methods Europe (FME '02) », série Lecture Notes in Computer Science, numéro 2391, Springer-Verlag, éditeurs L.-H. ERIKSSON, P. LINDSAY., pages 272-289, 2002.

[21] L. CHICLI, L. POTTIER, C. SIMPSON. *Mathematical quotients and quotient types in Coq.* in « TYPES Workshop », submitted to LNCS, Berg en Dal, Netherlands, 2002.

[22] P. COURTIEU. *Proving self-stabilization with a proof assistant.* in « Proceedings of IPDPS'2002 », série IEEE CS Press, 2002.

[23] M. HUISMAN. *Verification of Java's AbstractCollection class : a case study.* in « Mathematics of Program Construction (MPC'02) », série Lecture Notes in Computer Science, numéro 2386, Springer-Verlag, éditeurs E. BOITEN, B. MÖLLER., pages 175 - 194, 2002.

[24] A. IMINE, Y. SLIMANI, S. STRATULAT. *Inductive Theorem Prover Based Verification of Concurrent Algorithms.* in « MCSEAI02 (7th Maghrebian Conference on Computer Science) », volume 2, pages 313-324, 2002.

[25] A. IMINE, Y. SLIMANI, S. STRATULAT. *Using Automated Induction-based Theorem Provers for Reasoning on Concurrent Systems.* in « JFPLC'2002 (Onzièmes Journées Francophones de Programmation Logique et Programmation par Contraintes) », pages 71-85, 2002.

[26] A. MAHBOUBI, L. POTTIER. *Elimination des quantificateurs sur les réels en Coq.* in « Proceedings of JFLA'02 », éditeurs L. RIDEAU., 2002.

[27] H. NACIRI, L. RIDEAU. *Affichage et diffusion sur Internet d'explications en langue arabe de preuves mathématiques.* in « CARI'2002 6ème Colloque Africain sur la Recherche en Informatique », October, 2002.

[28] H. NACIRI, L. RIDEAU. *Formal Mathematical Proof Explanations in Natural Language Using MathML : An Application to Proofs in Arabic.* in « MathML International Conference 2002 », June, 2002.

[29] K. TRENTELMAN, M. HUISMAN. *Extending JML Specifications with Temporal Logic.* in « Algebraic Methodology And Software Technology (AMAST '02) », série Lecture Notes in Computer Science, numéro 2422, Springer-Verlag, éditeurs H. KIRCHNER, C. RINGEISSEN., pages 334-348, 2002.

Rapports de recherche et publications internes

[30] G. BARTHE, P. COURTIEU, G. DUFAY, M. HUISMAN, S. M. DE SOUSA, G. CHUGUNOV, L.-Å. FREDLUND, D. GUROV. *Temporal logic and toolset for applet verification : Compositional reasoning, model checking, abstract interpretation.* VerifiCard Deliverable 4.1., 2002.

[31] F. GUILHOT. *Proofs with Coq of theorems in plane geometry using oriented angles.* Rapport de recherche, numéro RR-4356, INRIA, 2002, http://www.inria.fr/rrrt/rr-4356.html.

[32] L. THÉRY. *A Tour of Formal Verification with Coq : Knuth's Algorithm for Prime Numbers.* Rapport de recherche, numéro RR-4600, INRIA, 2002, http://www.inria.fr/rrrt/rr-4600.html.

Divers

[33] Y. BERTOT, P. CASTÉRAN. *Le Coq'Art.* 2002, http://www-sop.inria.fr/lemme/Yves.Bertot/coqart.html, book to appear.