INRIA

INSTITUT NATIONAL DE RECHERCHE EN INFORMATIQUE ET EN AUTOMATIQUE

Avant-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 [Chargé de recherche INRIA]

Assistante de projet

Nathalie Bellesso

Personnel Inria

Francis Montagnac [Ingénieur de Recherche INRIA]
Laurence Rideau [Chargée de recherche INRIA, CDRI Sophia]
Laurent Théry [Chargé de recherche INRIA]

Collaborateurs extérieurs

Frédérique Guilhot [Professeur agrégé]
André Hirschowitz [Professeur UNSA, Laboratoire J.A.Dieudonné]
Roger Marlin [Professeur UNSA, Laboratoire J.A.Dieudonné]

Ingénieurs experts

Pascal Lequang
Claude Pasquier [depuis juillet 99]

Chercheurs doctorants

Antonia Balaa [bourse INRIA]
Laurent Chicli [bourse INRIA]
Yann Coscoy [bourse Dret jusqu'en mai 99, employé Caisse des dépôts ensuite]
Stéphane Lavirotte [bourse MESR]
Hanane Naciri [bourse MESR]
Olivier Pons [ATER CNAM puis Evry]

Chercheurs post-doctorants

Daniel Hirschkoff [ARC CFC mars-août 99]
Henrik Persson [ARC CFC depuis septembre 99]

Stagiaires

Jean Familiari [Maîtrise MIM UNSA, printemps 99]
Benoît Frondas [DEA MDFI, mai-juillet 99]
Virgile Prevosto [X 3^{e} année, juin-juillet 99]
Jérôme Waldispuhl [Maîtrise MIM UNSA, printemps 99]

2 Présentation et objectifs généraux

Les méthodes formelles prennent une place de plus en plus grande dans le développement de logiciels. L'objectif du projet Lemme est de contribuer à l'application de ces méthodes à la construction de logiciels pour le calcul scientifique.

Notre approche est originale en calcul scientifique en ce que nous ne cherchons pas à prouver des programmes (au sens de programmes écrits dans un langage conventionnel comme `C`, `Java`, ou `Caml`), mais au contraire à produire des programmes conformes à leurs spécifications **à partir de descriptions mathématiques "usuelles" des données, des algorithmes, et de leurs propriétés, ainsi que de leurs preuves**.

Nous cherchons à réduire la distance encore trop grande entre la résolution d'un problème mathématique (au sens large) sur le papier (ou au tableau) et les calculs sur ordinateur qu'il nécessite, en insistant sur la cohérence logique et le confort de ce passage de la théorie à la pratique.

Pour atteindre cet objectif, nous travaillons sur trois thèmes :

- le premier thème se concentre sur l'environnement de travail du chercheur, de l'ingénieur ou de l'étudiant qui développe des algorithmes certifiés et les démonstrations mathématiques nécessaires à la certification,
- le deuxième thème se concentre sur la formalisation de théories mathématiques pour décrire les objets de base de la connaissance scientifique,
- le troisième thème s'intéresse aux outils permettant de faciliter les démonstrations mathématiques et la production d'implémentations efficaces à partir des descriptions formelles des algorithmes et des démonstrations de leur correction.

Bien que pouvant se développer indépendemment, ces thèmes contribuer les uns aux autres de manière circulaire: l'environnement de travail du chercheur repose sur les outils de démonstration qui doivent être efficaces et sûrs. Le développement de ces outils de démonstration passe par la description certifiée d'algorithmes, qui repose elle-même sur une formalisation des outils mathématiques de base. Enfin, la formalisation des outils mathématiques de base est améliorée grâce à l'utilisation d'un environnement de travail pratique et efficace.

3 Fondements scientifiques

3.1 Environnements de preuves

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

Participants : Yves Bertot, Yann Coscoy, Stéphane Lavirotte, Pascal Lequang, Francis Montagnac, Hanane Naciri, Claude Pasquier, Olivier Pons, Loïc Pottier, Laurence Rideau, Laurent Théry.

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.

Deux objectifs complémentaires apparaissent. Le premier est d'améliorer les moyens d'interaction entre l'ordinateur et l'utilisateur, afin de rendre les outils accessibles à des ingénieurs peu intéressés par la théorie de la preuve et à des mathématiciens peu intéressés par la programmation et les contraintes formelles. Le second objectif est de faciliter la maintenance de grands développements de mathématiques formelles, afin d'aider leur réutilisation et leur adaptation à des contextes variés.

Ainsi, nous voulons augmenter la pénétration des méthodes formelles dans le développement des logiciels à la fois en facilitant leur usage par des débutants et en augmentant la productivité des utilisateurs confirmés.

3.1.1 Moyens d'interaction

En abordant le domaine des interfaces utilisateurs pour systèmes de preuve avec les outils initialement développés pour les environnements de programmation [TBK92], le projet CROAP, dont Lemme est issu, a permis une évolution rapide de ce domaine. Avec des concepts comme la preuve par sélection (*proof-by-pointing* [1]), il a acquis une grande notoriété.

La notion de preuve par sélection montre qu'une interaction riche entre l'utilisateur et la machine exige le développement de véritables compilateurs dont les données d'entrées forment une description symbolique des actions de l'utilisateur et dont les sorties sont des commandes exprimées dans le langage de bas niveau du système de preuve. De tels compilateurs méritent une description formelle précise et une réflexion sur les optimisations qui peuvent être appliquées sur le code engendré [2],[Ber97b].

Les outils d'interactions doivent bien sûr s'adapter aux domaines mathématiques rencontrés. L'expérience a montré que l'outil de preuve par sélection devenait inefficace lorsque l'on aborde les raisonnements algébriques. Nous avons mis à jour d'autres possibilités d'interaction efficace, fondées non seulement sur l'interprétation de positions désignées par l'utilisateur mais aussi sur l'interprétation de ses mouvements [Ber97a,Ber98b] et nous avons montré comment on pouvait rendre de tels outils facilement extensibles par l'utilisateur, pour s'adapter à des domaines mathématiques variés. Un défi important dans ce domaine de travail est de comprendre si les deux modèles d'interaction peuvent être joints de façon naturelle, de manière à fournir une interface utilisateur avec un apprentissage court mais un apport réel pour les débutants et les utilisateurs experts.

L'interprétation des actions de l'utilisateur avec la souris prend en charge les problèmes de communication de l'utilisateur vers la machine. Il est aussi important de maîtriser la communication de la machine vers l'utilisateur. Nous nous appuyons sur les recherches effectuées ces dernières années dans l'équipe CROAP sur l'outil d'affichage bi-dimensionnel, incrémental, et

[TBK92] L. Théry, Y. Bertot, G. Kahn, « Real Theorem Provers Deserve Real User-Interfaces », *Software Engineering Notes 17*, 5, 1992, p. 120–129, 5th Symposium on Software Development Environments.

[Ber97b] Y. Bertot, « Head-Tactics Simplification », *in: Algebraic Methodology and Software Technology, Lecture Notes in Computer Science, 1349*, Springer Verlag, Sydney, Australie, 1997.

[Ber97a] Y. Bertot, « Direct Manipulation of Algebraic Formulae in Interactive Proof Systems », *in: Electronic proceedings for the conference UITP'97*, Sophia Antipolis, septembre 1997, `http://www.inria.fr/croap/events/uitp97-papers.html`.

[Ber98b] Y. Bertot, « The CtCoq System: Design and Architecture », *Rapport de Recherche n° RR-3540*, INRIA, 1998, `http://www.inria.fr/RRRT/RR-3540.html`.

multi-thread `Figue` et son extension `Aïoli` pour la manipulation d'objets arborescents. Nous voulons compléter cette investigation pour mieux comprendre comment cet outil interagit avec les outils de visualisation existants ou en cours de développement pour le Web.

3.1.2 Maintenance des développements formels

Dans notre stratégie, il est nécessaire de prendre en compte non seulement les travaux nécessaires à la production de nouveaux logiciels certifiés et de nouveaux développements de mathématiques formelles mais également l'évolution sur le long terme de ces logiciels et développements. Par rapport à un développement mathématique traditionnel, un développement formel est plus sensible à des changements de l'environnement de travail : ce développement est construit sur une version précise de formalisation d'une théorie mathématique, utilisant les règles de raisonnement permises par un système formel précis. Notre travail sur la maintenance des développements formels vise à diminuer la fragilité des développements en facilitant leur adaptation d'un contexte à l'autre.

Pour améliorer l'utilisation des méthodes formelles dans l'industrie du logiciel, il faut résoudre les problèmes de maintenance qu'elles posent. En effet, lorsque les techniques formelles seront plus répandues, les développements verront leur taille augmenter et les préoccupations des développeurs (ingénieurs ou mathématiciens) évolueront d'un cadre logique vers un cadre logistique. Les grands développements formels seront comme de grandes librairies de calcul : ils devront être organisés comme tels, et les outils pour assurer leur cohérence restent à développer.

Pour apporter des réponses satisfaisantes dans ce domaine, nous pouvons organiser nos investigations autour des objectifs secondaires suivants :

Disposer de structures pour organiser les développements formels à un niveau d'abstraction plus élevé que les simples théorèmes. Ces travaux sont déjà abordés par l'étude des modules effectués dans l'équipe Coq. Cet objectif implique un besoin de comprendre les outils qui peuvent faciliter la maintenance de grands développements à partir de données organisées en modules.

Étudier spécifiquement les opérations nécessaires pour adapter des développements existants à des domaines d'applications légèrement différents. Ces travaux passent par une étude des dépendances entre objets mathématiques et la propagation de modifications le long des chaînes de dépendance.

Apporter un support pour les activités en amont et en aval d'un développement formel : en amont, description formelle ou informelle de modèles abstraits, raffinement de ces modèles, mise en correspondance et réutilisation des preuves; en aval, ré-ingéniérie de développements formels, documentation a posteriori, éventuellement par extraction systématique d'informations des développements formels (voir par exemple les travaux de Gilles Kahn, Laurent Théry et Yann Coscoy dans ce domaine [CKT95]), aide à l'isolation de concepts apparus pendant le travail de formalisation, mécanisation des opérations systématiques.

[CKT95] Y. Coscoy, G. Kahn, L. Théry, « Extracting Text from Proof », *in : Proceedings of "Typed Lambda-Calculi and Applications" (TLCA '95), LNCS, 905*, Springer-Verlag, 1995, Edimbourg, avril 1995.

Nous disposons déjà de plusieurs grands développements en `Coq`, l'algorithme de Buchberger certifié [6], l'algorithme de factorisation des polynômes [Mau98], l'algorithme de Stålmark pour la satisfiabilité des formules logiques du premier ordre [Let98], un compilateur [Ber98a], et une formalisation de l'algèbre élémentaire [15]. Nous serons à même de mesurer la validité de notre approche en évaluant la façon dont ces développements vieillissent et s'adaptent aux versions successives de `Coq`. Pour pousser nos investigations dans des conditions plus sélectives, nous pourrons également étudier la façon dont ces développements s'adaptent à d'autres outils de formalisation comme le système `NuPrl` de Cornell University.

3.2 Formalisation de théories mathématiques

Mots clés : formalisation, mathématiques.

Participants : Yves Bertot, Laurent Chicli, Frédérique Guilhot, André Hirschowitz, Loïc Pottier, Laurence Rideau.

Le but de ce thème est 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.

Ce thème est très lié aux deux autres, en ce sens qu'il doit fournir des fondements sémantiques aux objets des environnements de preuves, et des bibliothèques d'objets, lemmes et théorèmes de base pour alléger le travail de preuve des algorithmes mathématiques, dont la preuve nécessite souvent de connaître et de manipuler de larges pans des mathématiques classiques.

L'exemple qu'on a commencé à étudier est l'algèbre commutative : on s'est inspiré pour cela des formalisations en Coq de la théorie des ensembles et de la théorie des catégories. Par la suite viendront la géométrie algébrique, les théories des groupes et des corps finis, et la géométrie des polytopes (2D et 3D).

Le CCI est assez puissant pour formaliser des mathématiques complexes : structures algébriques avec leurs dépendances et leurs opérations (comme le système de calcul formel Axiom l'a fait), mais aussi, et surtout puisqu'on veut faire des démonstrations, toutes leurs propriétés logiques, ce qui reste très marginal dans les systèmes de calcul scientifique. Le CCI permet aussi d'écrire des algorithmes, sous forme de programmes fonctionnels récursifs, avec des structures de données riches. Enfin, avec l'isomorphisme de Curry-Howard, le CCI est un langage de manipulation des preuves mathématiques. Mais ce langage est difficile d'accès, un peu comme un assembleur, bien que sa concision et sa puissance le rendent très élégant pour l'initié.

Le système Coq, l'interface CtCoq offrent des outils qui permettent souvent une formalisation satisfaisante : les "record", les coercions, le pretty-printer PPML, la syntaxe extensible

[Mau98] F. Maurel, « Factorisation de polynômes à coefficients dans un corps fini », *Stage de première année*, Ens Ulm, Septembre 1998.

[Let98] P. Letouzey, « Algorithme certifié de Stålmarck », *Stage de première année*, Ens Ulm, Septembre 1998, `http://www.eleves.ens.fr:8080/home/letouzey/rapport.ps.gz`.

[Ber98a] Y. Bertot, « A certified compiler for an imperative language », *Rapport de Recherche n° RR-3488*, INRIA, 1998, `http://www.inria.fr/RRRT/RR-3488.html`.

de Coq par exemple. Mais de nombreux problèmes restent à résoudre : héritage multiple, complexité de l'empilement des coercions et du typage entre structures algébriques, manipulation difficile des types dépendants dans les preuves et les constructions d'objets, absence de sous-typage et de quotient dans le CCI.

On a pu à ce jour formaliser l'algèbre commutative et la théorie des catégories de base sans trop de problèmes [pag]. Il est néanmoins certain que de nouveaux outils auront à être développés et intégrés à `Coq` et `CtCoq` à l'avenir (tactiques dédiées, macros de génération de code `Coq`, algorithmes de réflexion, par exemple).

Plus fondamentalement, on étudie comment les types inductifs particuliers qu'on utilise pour représenter les structures mathématiques (record, ou Σ-types) peuvent être abstraits dans `Coq` lui-même pour permettre de calculer et raisonner sur eux à un niveau plus général (utiliser la notion de télescopes [Bru91] et les opérations de coercions associées, par exemple).

3.3 Implémentations certifiées d'algorithmes de calcul scientifique

Mots clés : algorithme, certification.

Participants : Antonia Balaa, Yves Bertot, Roger Marlin, Henrik Persson, Loïc Pottier, Laurent Théry.

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.

Par exemple, pour obtenir un programme de calcul de bases de Gröbner, on commence par écrire l'algorithme de Buchberger dans le CCI, puis on prouve en `Coq` qu'il calcule bien une base de Gröbner, enfin on utilise le processus d'extraction automatique de `Coq` pour produire un programme `Ocaml`, dont on sait qu'il calcule la même chose, puisque ceci est garanti par les propriétés du CCI et de son affaiblissement dans la théorie des types ML [6]. Le programme extrait est efficace, comme on pouvait le prévoir, et comme le montre l'expérience.

Cette approche a plusieurs avantages : on travaille à un haut niveau d'abstraction, en restant indépendant du langage d'implantation final (mais plus ce langage sera fonctionnel, plus l'implantation sera efficace). On peut aussi se concentrer sur les caractéristiques propres à l'algorithme qu'on étudie, en faisant abstraction du reste (gestion de la mémoire, choix d'implantation des données). Le niveau de certification peut être ainsi variable, et paramétrable (par exemple pour le calcul de bases de Gröbner, on peut supposer qu'une implantation de l'arithmétique des polynômes est donnée).

Mais elle pose aussi quelques problèmes qui, sans être bloquants, s'avèrent difficiles à résoudre. Par exemple il est très difficile actuellement de faire des preuves sur des algorithmes

[pag] « Action Coopérative "Calcul Formel Certifié" », INRIA, `http://www.inria.fr/croap/CFC/index.html`.

[Bru91] N. G. D. Bruijn, « Telescopic Mappings in Typed Lambda Calculus », *Information and Computation 91*, 2, 1991, p. 189–204.

récursifs dont la terminaison découle d'un ordre bien fondé[1]. Il est aussi délicat de travailler dans un style impératif, ou avec des opérations qui font des effets de bord (opérations en place sur les données par exemple).

Nous avons acquis une expérience originale dans ce domaine de la certification d'algorithmes mathématiques, où les expériences que nous avons menées ont toutes été concluantes : arithmétique des entiers longs, arithmétique des polynômes (multiplication de Karatzuba)[Pot97], algorithme de Buchberger [5], factorisation de polynômes [Mau98], satisfiabilité de formules booléennes [Let98].

L'efficacité des implantations obtenues est très bonne, puisqu'elle est au niveau de certains logiciels parmi les plus utilisés (`Maple`, `Mupad`, `Pari`).

Parmi les problèmes qui restent à résoudre, nous comptons traiter celui de l'extraction vers d'autres langages que `Ocaml` (quelques expériences avec C et `Java` sont encourageantes, et extraire vers `Maple` devrait être similaire), ainsi que celui de l'extraction de tout le CCI. En effet, actuellement, seule une partie du CCI (les sortes Set et Prop) s'extrait correctement en `Ocaml`. Nous avons une première implantation de cette extraction totale (incluant les univers de la sorte Type), vers `Ocaml` non typé, que nous comptons étendre et fonder théoriquement.

Notre but dans ce thème est triple : développer des outils nouveaux (extraction, récursion bien fondée), les valider sur des algorithmes fondamentaux du calcul scientifique, dont on testera les implantations sur des applications significatives.

4 Domaines d'applications

4.1 Télécommunications

Depuis les accidents catastrophiques des réseaux de commutateurs téléphoniques aux États-Unis, les opérateurs de télécommunication s'intéressent de façon croissante à l'utilisation de méthodes formelles dans la conception et la vérification des logiciels qui contrôlent leurs appareils. Bien sûr, une grande partie de ces méthodes formelles repose sur des outils d'exploration d'espaces finis d'état et de model-checking et font souvent appel à une étude théorique de la concurrence que nous n'abordons pas, mais les outils comparables à Coq sont aussi utilisés pour certains problèmes. Par exemple, les laboratoires Bell de Lucent Technologies accueillent une équipe qui réunit des utilisateurs de HOL (Elsa Gunter), Coq et CtCoq (Amy Felty), NuPrl (Doug Howe).

En France, nous avons des contacts avec l'équipe de Jean-François Monin au CNET de Lannion, qui explore depuis plusieurs années les différentes méthodes formelles applicables aux problèmes des opérateurs de télécommunication. Ces contacts ont donné lieu à une CTI (consultation technique informelle) du CNET, en collaboration avec le projet Coq. Ceci nous a permis

1. C'est le sujet de thèse d'Antonia Balaa, dirigée par Yves Bertot.

[Pot97] L. Pottier, « Certification d'algorithmes en Coq », *Séminaire ens-lyon*, INRIA, juin 1997.

[Mau98] F. Maurel, « Factorisation de polynômes à coefficients dans un corps fini », *Stage de première année*, Ens Ulm, Septembre 1998.

[Let98] P. Letouzey, « Algorithme certifié de Stålmarck », *Stage de première année*, Ens Ulm, Septembre 1998, `http://www.eleves.ens.fr:8080/home/letouzey/rapport.ps.gz`.

de financer la thèse d'Olivier Pons. Nous comptons poursuivre à l'avenir cette collaboration avec le CNET.

4.2 Cryptographie

La cryptographie est un domaine d'application privilégié pour les méthodes formelles. Pour l'instant, les méthodes formelles ont été utilisées pour prouver des protocoles [Bol97]. Nous avons engagé une réflexion avec le projet CODES pour comprendre comment la technique de certification que nous proposons pourrait s'appliquer aux briques de base de la cryptographie que sont les algorithmes de codage comme RSA (et ceux utilisés dans PGP par exemple).

4.3 Enseignement

Après les calculatrices et les langages de programmation structurée et typée, les systèmes de calcul formel ont fait leur entrée dans l'enseignement scientifique de premier cycle. Les systèmes permettant d'effectuer des démonstrations mathématiques vont les suivre logiquement, lorsqu'ils seront plus largement accessibles. Nous avons l'expérience depuis 3 ans de l'utilisation de Coq et CtCoq en maîtrise (Nice) et en DEA de mathématiques (MDFI). Comme on peut se passer de connaître et comprendre l'algorithme de Rish pour utiliser Maple, on peut aussi utiliser Coq sans connaître la théorie des types. Mieux, l'utilisation de Coq et CtCoq permet d'introduire par l'exemple les bases de la logique mathématique, qui ne sont pas enseignées comme telles, et que les étudiants apprennent (ou souvent n'apprennent pas) sur le tas.

Avec André Hirshowitz et Xiao Gang (Professeurs à l'Université de Nice-Sophia Antipolis, Laboratoire J. A. Dieudonné), et Frédérique Guilhot (Professeur agrégé en terminale), nous sommes convaincus que l'utilisation d'un système accessible de preuves formelles peut aider à montrer aux étudiants que les mathématiques élémentaires sont suffisamment simples et cohérentes pour être écrites et manipulées par une machine, qui d'habitude ne traite principalement que des nombres entiers, flottants ou des caractères. Cela peut aussi leur faire prendre conscience qu'une démonstration n'est pas une opération magique mais peut être décomposée en étapes simples selon des règles précises. Collaborer avec des enseignants devrait aussi nous permettre de confronter nos outils avec une large population d'utilisateurs novices, pour améliorer leurs fonctionnalités, leur ergonomie, etc.

5 Logiciels

5.1 Ctcoq

Participants : Yves Bertot [correspondant], Olivier Pons, Laurence Rideau.

Ctcoq (`http://www-sop.inria.fr/croap/ctcoq/ctcoq-fra.html`) continue à être distribué et à suivre les différentes versions de Coq, mais est appelé à disparaître, au profit de Pcoq, qui sera sa réincarnation en Java, au début du prochain millénaire.

[Bol97] D. Bolignano, « Towards a Mechanization of Cryptographic Protocol Verification », *in : Proceedings of the international conference on Computer-Aided Verification (CAV'97)*, juin 1997.

5.2 Aïoli et Figue

Participants : Claude Pasquier, 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 briques de base de Pcoq (le portage de Ctcoq en Java), et à ce titre ont reçu des améliorations: sélections multiples, nouveaux combinateurs 2D (matrices, crochets, fractions, racines n-ièmes).

5.3 Gbcoq

Participants : Loïc Pottier, Laurent Théry [correspondant].

Gbcoq (`http://www-sop.inria.fr/croap/CFC/Gbcoq.html`) est une implémentation de l'algorithme de Buchberger complètement certifiée.

6 Résultats nouveaux

6.1 Environnements de preuve

6.1.1 Portage des outils de preuve vers le langage Java

Participants : Yves Bertot, Pascal Lequang, Hanane Naciri, Loïc Pottier, Laurence Rideau.

Depuis plusieurs années, nous disposons d'un environnement de preuve pour le système Coq, basé sur le générateur d'environnements de programmation Centaur. Dans le courant de l'année 1999, nous avons adapté cet environnement de preuve en dehors de Centaur, en nous fondant cette fois-ci sur le langage Java. Pour marquer le changement de technologie, nous avons choisi de donner un nouveau nom à ce développement : Pcoq. Ce travail de re-développement nous permet d'isoler les caractéristiques centrales de l'environnement qui sont indépendantes du cadre fourni par Centaur. À plus long terme, nous espérons tirer de cette adaptation une meilleure distribution de nos outils et par conséquent une évaluation des idées d'interaction que nous expérimentons par une communauté plus large.

En outre, la mise en œuvre de Pcoq a servi de premier banc d'essai pour les librairies Figue et Aïoli qui ont été développée ces dernières années pour permettre la manipulation de données structurées comme les programmes où les formules mathématiques.

Parmi les fonctionnalités apportées à Pcoq, citons :

- les rattrapages d'erreurs de syntaxe, récupération et accrochage des commentaires,
- la mise à niveau de tous les pprinters,
- l'écriture de nouveaux combinateurs de Figue pour l'affichage des affichage de mathématiques (racines, sigma, etc).

Enfin la thèse d'Hanane Naciri (dirigée par Laurence Rideau) a débuté en octobre 99 avec comme sujet: « Conception et Réalisation d'outils pour l'interface homme-machine des environnements pour les mathématiques ».

Cette thèse se place dans le cadre du développement d'un environnement de travail pour la production de logiciels et de leurs démonstrations mathématiques par des méthodes formelles, s'appuie en particulier sur des outils améliorant les moyens d'interaction entre l'ordinateur et l'utilisateur. Dans ce cadre, nous nous appuyons sur les recherches effectuées ces dernières années dans l'équipe CROAP, puis l'équipe LEMME, sur l'outil d'affichage bi-dimensionnel, incrémental, et multi-thread Figue et son extension Aïoli pour la manipulation d'objets arborescents. Cette thèse devrait permettre d'offrir à l'utilisateur une panoplie d'outils nouveaux, en particulier des outils de débogage de l'affichage (déterminer quelle règle a affiché une expression donnée par exemple) et des outils d'aide à la création de nouvelles règles et de nouvelles notations.

6.1.2 Étude de l'intégration des standards de manipulation de données structurées dans les modules Figue et Aïoli.

Mots clés :
XML, XSL, MathML, DOM, édition structurée
.

Participant : Claude Pasquier.

La manipulation des données structurées constitue un axe de recherche important du projet. Des développement importants ont en particulier été effectués sur les modules Aïoli (manipulation de données structurées) et Figue (affichage bi-dimensionnel et incrémental). Ces outils reposent entièrement sur des technologies développées à l'INRIA mais sont conceptuellement très proches des standards de manipulation de documents structurés proposés par le W3C. Ainsi, XML (Extensible Markup Language) peut être utilisé pour représenter de manière standard des données structurées, DOM (Document Object Model) peut être considéré comme un équivalent du VTP, XSL (Extensible Stylesheet Language) comme un langage de script très proche de PPML et MathML comme un moyen d'exprimer le rendu de formules mathématiques. Le rapprochement de nos outils vers ces standards est évidemment souhaitable.

Un état de l'art relatif aux technologies XML a été réalisé en étudiant les différents standards existants (ou en cours d'élaboration) et en testant la plupart des outils XML non commerciaux.

La comparaison entre VTP et DOM a été développée dans un document intitulé « Représentation du package VTP en XML ». Une implémentation en DOM de la structure d'arbre manipulée par le VTP a été réalisée et a été testée avec Pcoq.

Une étude comparative entre PPML et XSL a été détaillée dans un document interne intitulé « PPML vs XSL ». Il ressort de cette étude qu'une partie importante des spécifications PPML peuvent être exprimées en XSL. Une illustration des principes de conversion a été effectuée en traduisant les prettyprinters des langages Java et Metal en XSL.

Nous pensons que, dans un premier temps, XML peut être utilisé comme un format d'échange et de stockage standard des données. La structure arborescente du VTP ainsi que les boîtes Figue disposent maintenant d'un format d'import/export en XML. Les DTD (Document Type Definition) associées sont obtenues automatiquement à partir du formalisme du langage. La sauvegarde d'une présentation Figue en XML a été ajoutée à Pcoq.

Une expérimentation est actuellement effectuée pour permettre à notre parseur PPML de travailler directement sur des documents XML.

6.1.3 Interacteur graphique structuré Aïoli

Participant : Laurent Théry.

Un travail de consolidation a été fait autour d'Aïoli. Une première version a été utilisée en interne à l'INRIA dans les projets OASIS, TROPICS et LANDE. Aïoli sert aussi comme brique de base pour la construction de l'interface de PCoq. Aïoli a été déposé a l'APP en juin 1999.

6.1.4 Reconnaissance de formules mathématiques

Participants : Stéphane Lavirotte, Loïc Pottier.

Stéphane Lavirotte termine la rédaction de son mémoire de thèse sur la reconnaissance de formules mathématiques. La méthode élaborée les années passées n'a pas subi de changement majeur. Diverses améliorations ont été apportées, surtout algorithmiques, pour la rendre plus rapide. Le travail en commun avec Andréa Kosmala a été poursuivi et doit conduire fin 99 à un prototype de saisie manuelle par tablette graphique et de reconnaissance de formules mathématiques mono-utilisateur. Des extensions de la grammaire utilisée sont en cours (matrices, diagrammes). Des connexions à OpenMath et à Pcoq sont prévues.

6.1.5 Wims

Participant : Loïc Pottier.

Wims (`http://wims.unice.fr`) est un serveur d'exercices mathématiques sur le Web développé par Gang Xiao, professeur au laboratoire J.A. Dieudonné de l'UNSA. On a développé dans ce cadre une interface à Coq, permettant à un utilisateur non averti d'effectuer des preuves de formules logiques avec Coq. Cette interface a été utilisée en TD dans la maîtrise Math-Info de l'UNSA. Techniquement cette réalisation a permis de tester l'utilisation de Ocaml comme langage de prototypage rapide d'applications internet, et de tester Coq dans une utilisation non standard, où chaque session Coq ne dure que quelques fractions de seconde. Plus généralement, ce travail est un premier essai d'application à l'enseignement des résultats autour de Coq et de Ctcoq, aussi bien qu'un début de collaboration avec des enseignants en mathématiques férus d'informatique mais non-spécialistes de la théorie de la preuve.

6.1.6 Preuves textuelles

Participant : Loïc Pottier.

En s'inspirant du travail de Yann Coscoy (qui termine sa thèse), on a développé un nouveau module de génération de preuve textuelle en ocaml dans Coq. Ce module est intégré à la version courante de Pcoq, ainsi qu'à une version encore expérimentale de Wims. On ne se base plus sur le lambda-terme pour créer le texte d'une preuve, ni sur le script de preuve, mais sur

l'arbre des sous-buts et des tactiques utilisées. Pour l'instant la traduction est locale à l'arbre de preuve, mais à l'avenir elle pourra être globale (pour bien expliquer une preuve non triviale, il faut en avoir une vue globale). Dans ce cadre, on a développé des tactiques spécialisées. On a aussi ajouté dans Pcoq des fonctionnalités de « proof-by-pointing » dans le texte de preuve. Un objectif de travail est de faire complètement disparaître le script ésotérique des tactiques de Coq pour ne laisser que le texte de la preuve en cours de développement, sur lequel on agit, à la souris ou au clavier.

6.2 Formalisation de théories mathématiques.

6.2.1 Formalisation de nouvelles structures mathématiques en Coq.

Participant : Laurent Chicli.

Ce travail se fonde sur la formalisation de structures algébriques effectuée par Loïc Pottier en 98. Un index des notions formalisées se trouve a l'URL `http://www-sop.inria.fr/lemme/personnel/Laurent.Chicli/FrCoq.html`. Il s'agit principalement d'une part de notions topologiques (ouverts, fermés, compacts, applications continues etc..) et d'autre part de la théorie des faisceaux (préfaisceaux, faisceaux, germes, fibres et faisceaux associés), dont le développement est inspiré du livre 'Algebraic Geometry' de R. Hartshorne.

6.2.2 Formalisation des matrices en Coq

Participant : Frédérique Guilhot.

En s'inspirant du cours d'algèbre d'André Hirschowitz en DEUG à l'UNSA, et des livres de Serge Lang, on a formalisé en Coq les bases de la théorie des matrices. Un effort important a été fait pour que cette formalisation en Ctcoq soit lisible par un néophyte; ce qui a été possible grâce à PPML, du moins pour les définitions et les énoncés, mais pas encore pour les preuves qui restent sous forme de scripts de tactiques Coq. La suite de ce travail concerne maintenant les nombres complexes, toujours en se fondant sur le cours d'André Hirschowitz.

6.2.3 Télescopes

Participant : Loïc Pottier.

A partir d'une suggestion de N.G.De Bruijn[2] on a développé en Coq la notion de télescope, (dans un type inductif qui s'est avéré isomorphe à celui des ensembles de Peter Ackzel), et montré ses propriétés de base. Cette notion est sans doute la bonne pour représenter les structures mathématiques et pour raisonner sur elles, chose indispensables si on veut effectuer des preuves complexes, aussi bien que si on veut certifier des algorithmes de calcul formel. Ce travail a été repris et développé dans l'Action Coopérative de Recherche CFC par Sylvain Boulmé (article aux JFLA 99).

2. « Telescopic Mappings in Typed Lambda Calculus », N.G.De Bruijn, Information and Computation, Vol. 91, No. 2, April 91.

6.2.4 Records primitifs en Coq.

Participant : Daniel Hirschkoff.

Voir la section sur l'action coopérative CFC (8.1.1).

6.3 Certification d'algorithmes.

6.3.1 Récursion bien fondée dans Coq

Participants : Antonia Balaa, Yves Bertot.

Dans le cadre de son travail de thèse, Antonia Balaa étudie des outils pour faciliter le raisonnement sur le comportement des fonctions définies par récursion bien fondée. Pour disposer d'exemples significatifs utilisant de telles fonctions elle a également travaillé au développement d'une étude de cas dans l'algorithmique des graphes, en s'intéressant particulièrement au calcul d'arbres de recouvrement minimal. Ce travail, non terminé à l'heure de rédaction de ces lignes, permettra également d'améliorer nos connaissances sur la comparaison entre plusieurs méthodes formelles, puisque nous disposons également d'une description de l'algorithme de calcul d'arbre de recouvrement effectuée avec la méthode B il y a quelques années par un autre doctorant.

6.3.2 Comparaison de Coq avec la méthode B

Participant : Yves Bertot.

Dans le cadre d'une étude proposée par la société Gem+, nous avons entamé la formalisation en Coq d'un problème déjà étudié par eux en utilisant la méthode B. Cette étude est effectuée avec l'objectif de comprendre dans quelle mesure les types dépendants et les constructions inductives fournies dans Coq facilitent le travail de spécification et de vérification formelle d'algorithmes portant sur l'analyse de programmes. À titre scientifique, nous espérons tirer de cette étude une meilleure compréhension des outils nécessaires pour rendre faisable les objectifs de formalisation qui apparaissent dans ce genre de travaux.

6.3.3 Certification de l'algorithme de Stålmarck

Participant : Laurent Théry.

La preuve initiale sur l'algorithme de Stålmarck (qui permet de savoir rapidement en pratique si une formule booléenne est une tautologie) a été étendue pour introduire des optimisations. Ce travail a donne lieu à un développement conséquent sur les structures avec pointeur.

6.3.4 Étude de la complexité des fonctions en Coq

Participants : Benoît Frondas, Laurent Théry.

Dans le cadre de son stage de DEA Benoit Frondas a développé une méthode pour raisonner sur la complexité d'un programme à l'intérieur de Coq. L'idée est de faire correspondre à chaque

fonction en Coq une fonction qui calcule une mesure. Un premier prototype a été construit qui permet, étant donné la définition du tri par insertion, de montrer que sa complexité est en n^2.

6.3.5 Extraction de Coq en Java.

Participant : Virgile Prevosto.

Voir la section sur l'Action Coopérative de Recherche CFC (8.1.1).

6.4 Coefficients du polynome d'Ehrart et application à la reconnaissance de formes géométriques

Participants : Gérard Giraudon, Jean Familiari, Grégoire Malandain, Loïc Pottier, Jérôme Waldispuhl.

Le problème était, pour les deux stagiaires Jérôme Waldispuhl et Jean Familiari, de déterminer si les coefficients du polynôme d'Ehrart d'un polytope ou d'une forme générale (plan ou 3D) pouvaient être utilisés pour reconnaître ce polytope ou cette forme dans une image, avec l'idée d'appliquer cette étude à la reconnaissance de formes dans les images médicales (2D et 3D). Après une étude théorique des polynômes d'Ehrart de différentes formes simples, il s'est avéré que le polynôme d'Ehrart et ses coefficients n'avaient pas un comportement robuste vis à vis des déplacements dans l'espace (translations et rotations). L'étude s'est alors portée sur la poursuite de l'étude de la matrice de cooccurrence, qui permet de définir un autre polynôme associé à une forme. On a dressé une zoologie partielle de ces coefficients et polynômes pour des formes variées théoriques ou provenant d'images médicales 2D et 3D segmentées, permettant de faire apparaître des caractéristiques comme la compacité, l'angulosité, qui peuvent être calculées efficacement et utilisées ensuite dans la reconnaissance.

7 Contrats industriels (nationaux, européens et internationaux)

7.1 Dyade

Participant : Claude Pasquier.

Résumé :

Dyade fournit les ressources pour le contrat de Claude Pasquier sur l'adaptation de Figue et Aïoli aux standards XML.

7.2 GénieII

Participant : Pascal Lequang.

Résumé :

Génie fournit les ressources pour le contrat de Pascal Lequang sur le développement de `CtCoq` *et son portage en Java.*

7.3 Dassault Aviation

Participant : Antonia Balaa.

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

8 Actions régionales, nationales et internationales

8.1 Actions nationales

8.1.1 Action coopérative CFC

Participants : Antonia Balaa, Yves Bertot, Laurent Chicli, Daniel Hirsckoff, Henrik Persson, Loïc Pottier, Laurence Rideau, Laurent Théry.

L'année 1999 a été la deuxième année de l'action « Calcul Formel Certifié ». Pour notre part nous y avons contribué de différentes manières :

- développement de la notion de télescope en Coq, de la théorie associée, et application à la représentation des structures algébriques,
- différents exposés sur Ctcoq, Pcoq, nos développement de théories mathématiques et d'algorithmique en Coq, lors des réunions régulières à Paris.
- préparation de la réunion ouverte CFC en juin à Paris, avec des intervenants étrangers variés (Olga Caprotti),
- encadrement d'un stagiaire : Virgile Prevosto, en troisième année de l'École Polytechnique, est venu faire son stage à Sophia sur l'extraction de Coq vers Java. Il a effectué un travail concluant, qui montre que l'on peut obtenir du code Java certifié et relativement lisible à partir de code Coq (l'exemple traité concerne la cryptographie). Une des difficultés était de traduire les types inductifs de Coq en classes Java, ce qui a été en partie fait.
- travail avec Daniel Hirschkoff dans le cadre d'un post-doc CFC. Il s'agissait d'implémenter en Coq une notion de « record » primitifs, dans la veine des enregistrements extensifs d'Ocaml. Ce travail est initialisé, au moins dans la spécification théorique, en collaboration avec Gilles Barthe. L'implémentation devrait commencer bientôt, mais en dehors de CFC, Daniel Hirshkoff ayant été recruté à l'ENS Lyon.
- Henrik Persson, qui a fait sa thèse sous la direction de Thierry Coquand à Chalmers, a repris la suite du post-doc CFC en septembre. Son premier projet a été de modifier une formalisation par Laurent Théry dans Coq de l'algorithme de Buchberger. En adoptant un résultat de sa thèse de doctorat, il a obtenu une preuve constructive d'existence des bases de Gröbner dans Coq, à partir duquel une version récursive structurelle de l'algorithme de Buchberger a été extraite. Il attaque maintenant le problème de trouver des définitions catégoriques mais à l'extraction efficace pour les structures algébriques du calcul formel.

8.2 Actions européennes

8.2.1 Réseau Types

Lemme, à la suite de Croap, participe activement au réseau Types, lui-même suite du BRA Types. Ce réseau, dont c'est la dernière année, va se poursuivre sous des formes non encore déterminées. En tout état de cause, Lemme fera partie de la proposition de nouveau réseau, et de deux autres propositions européennes FET en cours de montage. Cette année, Yves Bertot et Laurent Théry ont fait des cours à l'École d'été Types début septembre. Francis Montagnac a assuré l'installation des machines dans cette École.

8.2.2 Réseau Calculemus

Cette année nous n'avons pas participé activement au projet Calculemus, mais avons continué à nous y intéresser. Loïc Pottier a assisté au workshop Calculemus à Trento en juillet.

8.3 Actions internationales

8.3.1 UITP (User-Interfaces for Theorem Provers)

Nous participons depuis sa création au cycle de colloques UITP où nous rencontrons la majeure partie des équipes dans le monde qui s'intéressent à ce domaine. Les objectifs des différentes équipes varient de l'outil de preuve pour l'enseignement de la logique à l'outil de démonstration automatique dans un thème particulier, en passant par l'étude générale de l'ergonomie des interfaces et les outils d'édition de documents mathématiques. Nos compétences dans ce forum sont reconnues, puisque nous avons été invités à plusieurs reprises dans le comité de programme et nous avons organisé l'édition 1997.

9 Diffusion de résultats

9.1 Animation de la communauté scientifique

Yves Bertot et Laurent Théry, en collaboration avec Christine Paulin et Gilles Dowek de l'équipe Coq à l'INRIA Rocquencourt et André Hirschowitz de l'Université de Nice, ont organisé la conférence TPHOLs'99 qui a eu lieu à Nice du 14 au 17 septembre 1999. Cette conférence a réuni 60 participants dont 25 orateurs. Les actes de la conférence ont été publiés chez Springer-Verlag dans la série « Lecture Notes in Computer Science » sous le numéro 1690. Des actes supplémentaires, réunissant des descriptions de travaux en cours ont été publiés par l'INRIA sous le titre: "TPHOLs'99, 12th International Conference Theorem Proving in Higher Order Logics: Emerging Trends".

Yves Bertot était membre du comité de programme pour cette conférence.

Yves Bertot a participé au jury de thèse de Philippe Ladagnous, thèse soutenue le 19 octobre 1999 à l'Université Paul Sabatier à Toulouse.

Loïc Pottier est promoteur de l'action coopérative « Calcul Formel Certifié ».

Loïc Pottier est membre nommé du CNU 27 section.

Laurence Rideau remplit la fonction de CDRI de l'UR de Sophia Antipolis.

Laurence Rideau et Yves Bertot ont participé au jury de la thèse d'Olivier Pons, soutenue au CNAM le 13 Juillet 1999.

Laurent Théry est promoteur de la demande d'action coopérative « Arithmétique des Ordinateurs Certifiée ».

9.2 Direction de thèses

Yves Bertot dirige la thèse d'Antonia Balaa.

Loïc Pottier dirige la thèse de Stéphane Lavirotte, et codirige avec André Hirschowitz celle de Laurent Chicli.

Laurence Rideau dirige la thèse de Hanane Naciri.

9.3 Enseignement

Comme l'année passée, Loïc Pottier et Laurent Théry ont donné un TD "preuves d'algorithmes en `Coq` et `CtCoq`", lors de la semaine intensive de novembre 99 à Sophia.

Laurent Théry a participé à une option du DEA de mathématique discrète de Marseille.

Yves Bertot a donné 10 heures de cours au DEA « Mathématiques discrètes et fondements de l'informatique » de Marseille.

Yves Bertot et Laurent Théry ont donné 6 heures de cours à l'ISIA.

Laurent Théry s'est rendu en visite pour une semaine à l'université de Minho en juin.

Yves Bertot et Laurent Théry ont donné un exposé à l'école Types en septembre.

Loïc Pottier a donné 52h de cours d'algorithmique et logique en maîtrise Math-Info à l'UNSA.

10 Bibliographie

Ouvrages et articles de référence de l'équipe

[1] Y. Bertot, G. Kahn, L. Théry, « Proof by Pointing », *in : Theoretical Aspects of Computer Software*, *Lecture Notes in Computer Science*, *789*, p. 141–160, Sendai, Japan, 1994.

[2] Y. Bertot, L. Théry, « A Generic Approach to Building User Interfaces for Theorem Provers », *Journal of Symbolic Computation 25*, 1998, p. 161–194.

[3] Y. Bertot, « The CtCoq System: Design and Architecture », *Rapport de Recherche n° RR-3540*, INRIA, 1998, `http://www.inria.fr/RRRT/RR-3540.html`.

[4] S. Lavirotte, L. Pottier, « Optical formula recognition », *in : "Proc. of International Conference on Document Analysis and Recognition (ICDAR'97)"*, Ulm, Allemagne, 1997.

[5] L. Pottier, L. Théry, « Certified Computer Algebra », *in : Calculemus workshop*, Eindhoven, juillet 1998. Extended Abstract.

[6] L. Théry, « A certified version of Buchberger's algorithm », *in : Automated Deduction—CADE-15*, *LNAI*, *1421*, Springer-Verlag, p. 349–364, 1998.

Thèses et habilitations à diriger des recherches

[7] Y. Bertot, *Des environnements de preuve aux environnements de programmation*, Habilitation à diriger des recherches, université de nice-sophia antipolis, octobre 1999.

[8] O. Pons, *Conception et réalisation d'outils d'aide au développement de grosses théories dans les systèmes de preuve interactifs*, thèse de doctorat, juillet 1999, Thèse d'Université, effectuée sous la responsabilité de Laurence Rideau et Yves Bertot.

Articles et chapitres de livre

[9] J. R. Harrison, L. Théry, « A Skeptic's Approach to Combining HOL and Maple », *Journal of Automated Reasoning 21*, 3, 1998, p. 295–325.

Communications à des congrès, colloques, etc.

[10] Y. Bertot, « Ctcoq et PCoq », *in : Poster presented at ISSAC'99, Vancouver, Canada*, juillet 1999.

[11] Y. Bertot, « Ctcoq et PCoq », *in : FM'99, Toulouse, colloque satellite des utilisateurs de Coq*, septembre 1999.

[12] S. Lavirotte, A. Kosmala, L. Pottier, G. Rigoll, « On-Line Handwritten Formula Recognition using Hidden Markov Models and Context Dependent Graph Grammars », *in : ICDAR'99. Proceeding of Fifth International Conference on Document analysis and Recognition. Bangalore, India, september 1999."*, septembre 1999.

[13] L. Pottier, « Formalization of algebra in Coq and Ctcoq », *in : Workshop Types, Lökeberg, Suède*, juin 1999.

[14] L. Pottier, « La contrib algebra de Coq », *in : FM'99, Toulouse, colloque satellite des utilisateurs de Coq*, septembre 1999.

Rapports de recherche et publications internes

[15] L. Pottier, « Un début de formalisation de l'algèbre en Coq et Ctcoq », *Rapport de Recherche n° à paraître*, INRIA, 1999, `http://www.inria.fr/RRRT/RR-????.html`.