Intégration d'outils de preuves automatiques à Coq : applications à la géométrie

Laboratoire d'accueil : ICube UMR 7357 CNRS - Université de Strasbourg

Directeurs de thèse : Nicolas Magaud (magaud@unistra.fr), Pascal Schreck (schreck@unistra.fr)

Mots-clés : Coq, tactiques, outils d'aide à la preuve, preuves automatiques

Les assistants de preuve comme Coq [5, 3] permettent, dans un même outil, non seulement d'implanter sous forme fonctionnelle des objets mathématiques et des opérations sur ces objets, mais aussi de démontrer formellement leurs propriétés. Ce travail de spécification et de preuve est confié à l'utilisateur, le système se contentant de vérifier, au fur et à mesure, que le raisonnement construit est bien correct. Cette approche permet de s'affranchir des limitations des outils automatiques et de tirer profit, à plein, de l'intelligence et de l'inventivité humaines. Néanmoins, plus les preuves sont complexes, et plus il est nécessaire de fournir des outils pratiques pour aider le développeur. Il existe bien sûr quelques procédures de décision pour des fragments décidables des théories considérées. Très utilisés en pratique, ces outils sont cependant bien moins puissants que certains prouveurs automatiques comme Vampire [8], CVC4 [2] ou Z3 [6]. En plus de ces prouveurs génériques, il existe également un certain nombre de prouveurs spécifiques, comme celui dédié à la géométrie projective, développé récemment dans l'équipe [4].

L'objectif de cette thèse sera d'étudier comment exploiter efficacement en Coq ces outils automatiques pour aider l'utilisateur dans son travail de preuve. Deux directions seront privilégiées.

Il s'agira dans un premier temps de faciliter la communication et l'interaction entre les prouveurs automatiques et Coq. Pour cela, les traces de preuves générées par les prouveurs automatiques devront être transformées en certificats vérifiables automatiquement par Coq. Cela pourra se faire en s'inspirant de l'approche proposée dans [1, 7]. Un premier exemple d'application consistera à rendre Coq et le prouveur basé sur la géométrie projective inter-opérable. L'outil automatique pourra alors être appelé à tout moment au sein d'une session de développement d'une preuve formelle et produira un fragment de script Coq directement applicable pour faire progresser la démonstration. Dans un deuxième temps, on cherchera à intégrer directement dans Coq certains mécanismes de raisonnement utilisés dans les prouveurs automatiques en les implantant sous forme de *plug-ins* dans Coq et en les rendant immédiatement disponibles sous forme de tactiques.

Afin d'évaluer les capacités et les performances des outils mis en oeuvre, nous nous intéresserons, en premier lieu, aux modèles finis de la géométrie projective où la taille des modèles est un obstacle à leur formalisation dans un système interactif comme Coq. D'autres applications pourront être envisagées, notamment en combinatoire des objets géométriques. Il s'agira, par exemple, d'énumérer certaines familles de cartes, d'établir des bijections entre des représentations non trivialement équivalentes des cartes ainsi qu'entre les cartes planaires enracinées et λ-termes planaires comme cela est présenté dans [9].

Références

[1] Michaël Armand, Germain Faure, Benjamin Grégoire, Chantal Keller, Laurent Théry, and Benjamin Werner. A modular integration of SAT/SMT solvers to coq through proof witnesses. In Jean-Pierre Jouannaud and Zhong Shao, editors, *Certified Programs and Proofs - First International Conference, CPP 2011, Kenting, Taiwan, December 7-9, 2011. Proceedings*, volume 7086 of *Lecture Notes in Computer Science*, pages 135–150. Springer, 2011.

[2] Clark W. Barrett, Christopher L. Conway, Morgan Deters, Liana Hadarean, Dejan Jovanovic, Tim King, Andrew Reynolds, and Cesare Tinelli. CVC4. In Ganesh Gopalakrishnan and Shaz Qadeer, editors, *Computer Aided Verification - 23rd International Conference, CAV 2011, Snowbird, UT,*

USA, July 14-20, 2011. Proceedings, volume 6806 of *Lecture Notes in Computer Science*, pages 171–177. Springer, 2011.

[3] Yves Bertot and Pierre Castéran. *Interactive Theorem Proving and Program Development, Coq'Art : The Calculus of Inductive Constructions*. Springer, 2004.

[4] David Braun. *Approche combinatoire pour l'automatisation en Coq des preuves formelles en géométrie d'incidence projective*. PhD thesis, Université de Strasbourg, sept. 2019.

[5] Coq development team. *The Coq Proof Assistant Reference Manual, Version 8.12.0*, 2020.

[6] Leonardo Mendonça de Moura and Nikolaj Bjørner. Z3 : an efficient SMT solver. In C. R. Ramakrishnan and Jakob Rehof, editors, *Tools and Algorithms for the Construction and Analysis of Systems, 14th International Conference, TACAS 2008, Held as Part of the Joint European Conferences on Theory and Practice of Software, ETAPS 2008, Budapest, Hungary, March 29-April 6, 2008. Proceedings*, volume 4963 of *Lecture Notes in Computer Science*, pages 337–340. Springer, 2008.

[7] Burak Ekici, Alain Mebsout, Cesare Tinelli, Chantal Keller, Guy Katz, Andrew Reynolds, and Clark W. Barrett. Smtcoq : A plug-in for integrating SMT solvers into coq. In Rupak Majumdar and Viktor Kuncak, editors, *Computer Aided Verification - 29th International Conference, CAV 2017, Heidelberg, Germany, July 24-28, 2017, Proceedings, Part II*, volume 10427 of *Lecture Notes in Computer Science*, pages 126–133. Springer, 2017.

[8] Alexandre Riazanov and Andrei Voronkov. The design and implementation of VAMPIRE. *AI Commun.*, 15(2-3) :91–110, 2002.

[9] Noam Zeilberger and Alain Giorgetti. A correspondence between rooted planar maps and normal planar lambda terms. *Logical Methods in Computer Science*, 11(3), 2015.