Proposition de stage

Utilisation de techniques de « machine learning » pour l'aide à la preuve dans le système Coq

Sujet

Les outils d'aide à la preuve sont à la base de l'ingénierie logicielle sûre. Klein et al. ont prouvé la correction du micro-noyau seL4 en Isabelle/HOL. Leroy a développé CompCert, un compilateur C vérifié formellement en utilisant Coq. Kumar et al. ont construit un compilateur vérifié formellement pour un langage de programmation fonctionnel, CakeML, en HOL4. En mathématiques, les mathématiciens ont remplacé leurs preuves sur papier par des preuves mécanisées pour éviter toute erreur humaine dans leurs preuves : Hales et al. ont prouvé mécaniquement la conjecture de Kepler en utilisant HOL Light et Isabelle/HOL, tandis que Gonthier et al. ont réalisé la preuve formelle du théorème des quatre couleurs en Coq. En informatique théorique, Paulson a démontré les théorèmes d'incomplétude de Gödel en utilisant Nominal Isabelle.

Pour faciliter le développement efficace de preuves dans de tels projets de vérification à grande échelle, les outils d'aide à la preuve modernes offrent un certain nombre d'outils, dont une panoplie de méthodes de preuve, appelées aussi tactiques. Par exemple, la bibliothèque standard d'Isabelle/HOL contient 160 tactiques. Ces ensembles de tactiques fournissent une automatisation très utile pour la démonstration interactive de théorèmes. Cependant, il faut toujours une expertise spécifique à l'outil d'aide à la preuve pour choisir la bonne tactique afin de décharger un but de preuve donné.

Dans ce stage, nous nous proposons de construire, dans le cadre du système Coq [3], une méthode d'aide à la preuve qui étant donné un but de preuve, pourrait recommander une tactique à appliquer. L'idée est d'extraire cette recommandation en appliquant des algorithmes de « machine learning » à des grands corpus de preuves (typiquement la bibliothèque standard de Coq, par exemple). Pour ce faire, on pourra s'inspirer de travaux récents réalisés dans le cadre d'Isabelle/HOL [2] ou de premières expérimentations qui ont été réalisées dans le cadre de Coq [1].

Parmi les problèmes à résoudre, il faudra notamment se poser la question de quelles caractéristiques on extraie des preuves et également quels algorithmes on utilise (on pourra utiliser plusieurs approches et les comparer). Comme Coq repose sur l'isomorphisme de Curry-Howard (ce qui lui permet d'encoder les preuves comme des fonctions), on pourra également regarder si cette structure particulière des preuves peut être exploitée avantageusement par les algorithmes de « machine learning ».

Il faudra également valider notre approche. Cela pourra se faire soit en utilisant toute la bibliothèque standard pour l'apprentissage et tester notre méthode sur les contributions utilisateurs de Coq (disponibles sur le site web de Coq), soit en utilisant une partie de la bibliothèque standard (par exemple, 90%) pour l'apprentissage et le reste de la bibliothèque standard (les 10% restants) pour tester notre méthode.

Travail à réaliser

— Caractérisation des structures de données représentant les états de preuve, qui seront manipulées par les algorithmes de « machine learning » ;
— Choix d'un algorithme de « machine learning », qui va exploiter les structures de données exhibées précédemment ;
— Implantation de la méthode de recommandation de tactique et validation sur un corpus et un ensemble de preuves à déterminer.

Remarques additionnelles

Le stage s'effectuera au sein de l'équipe MaREL du LIRMM. Le stage sera rémunéré. L'encadrement sera réalisé par :
— David Delahaye (Université de Montpellier, LIRMM, David.Delahaye@lirmm.fr).

Références

[1] C. Kaliszyk, L. Mamane, and J. Urban. Machine Learning of Coq Proof Guidance : First Experiments. In *Symbolic Computation in Software Science (SCSS)*, volume 30 of *EPiC Series in Computing*, pages 27–34, Gammarth (La Marsa, Tunisia), Dec. 2014. EasyChair.

[2] Y. Nagashima and Y. He. PaMpeR : Proof Method Recommendation System for Isabelle/HOL. In *Automated Software Engineering (ASE)*, pages 362–372, Montpellier (France), Sept. 2018. ACM.

[3] The Coq Development Team. *Coq, version 8.8.2*. Inria, Sept. 2018. http://coq.inria.fr/.