ECOLE
DOCTORALE

Titre : Inférence Profonde pour la Démonstration Graphique de Théorèmes

Mots clés : Assistants de preuve, Théorie de la démonstration, Interfaces humain-machine, Mathématiques formelles, Inférence profonde, Manipulation directe

Résumé : Les assistants de preuve sont des logiciels permettant de vérifier rigoureusement des raisonnements mathématiques. Ils peuvent être généraux (comme Coq, Lean, Isabelle...) ou plus spécialisés (comme EasyCrypt). Ils permettent un niveau de précision qui certifie qu'aucune erreur ne peut se produire, mais restent difficiles d'utilisation.
Nous proposons un nouveau paradigme de construction de preuves formelles par actions effectuées dans une interface graphique, afin de permettre une utilisation plus confortable et plus intuitive. Notre paradigme s'appuie sur des principes de manipulation directe, combinant des techniques d'interactions anciennes (Proof-by-Pointing) et récentes (Proof-by-Linking) qui exploitent les dernières avancées en théorie de la démonstration par inférence profonde. Nous implantons le paradigme dans un prototype d'interface graphique appelé Actema, que nous intégrons par la suite à l'assistant de preuves Coq en concevant le plugin coq-actema. Ce dernier offre aux utilisateurs de Coq la possibilité d'intégrer des preuves graphiques dans des développements textuels existants.
Puis nous explorons une série de systèmes d'inférence profonde qui donnent plus de structure à la notion de but logique. Ces systèmes ont en commun de pouvoir représenter les buts de deux manières alternatives : soit textuellement au travers d'une syntaxe inductive standard, soit graphiquement à l'aide d'une notation métaphorique adaptée à la manipulation directe.
La première famille de systèmes, appelée calculs de bulles, est une reformulation topologique de la théorie des séquents imbriqués. Elle permet un partage efficace des contextes communs entre sous-buts, facilitant la factorisation des étapes de raisonnement avant et arrière. Le second système, appelé calcul des fleurs, est un raffinement intuitionniste de la théorie des graphes existentiels de C. S. Peirce. Les deux types de systèmes sont démontrés analytiques et complètement réversibles, ce qui les rend adaptés à des techniques d'automatisation de la preuve. Nous développons finalement le Flower Prover, un prototype d'interface graphique pour la construction interactive de démonstrations basé sur le calcul des fleurs.

Title : Deep Inference for Graphical Theorem Proving

Keywords : Proof assistants, Proof theory, Human-computer interfaces, Formal mathematics, Deep inference, Direct manipulation

Abstract : Proof assistants are software systems that allow for the precise checking of mathematical reasoning. They can be general purpose (like Coq, Lean, Isabelle...) or more specialized like EasyCrypt. They enable a level of accuracy which certifies that no error can occur, but remain difficult to use.
We propose a new paradigm for constructing formal proofs through actions performed in a graphical user interface (GUI), in order to enable a more comfortable and intuitive use. Our paradigm builds upon direct manipulation principles, combining both old (Proof-by-Pointing) and new (Proof-by-Linking) interaction techniques that exploit recent advances in deep inference proof theory. We implement the paradigm in a prototype of GUI called Actema, which we subsequently integrate into the Coq proof assistant by designing the coq-actema plugin. The latter offers Coq users the ability to integrate graphical proofs into existing textual developments.
We then explore a series of deep inference proof systems that give more structure to the notion of logical goal. These systems share the ability to represent goals in two alternative ways: either textually through a standard inductive syntax, or graphically using a metaphorical notation well-suited to direct manipulation.
The first family of systems, called bubble calculi, is a topological reformulation of the theory of nested sequents. It allows for efficient sharing of hypotheses and conclusions among subgoals, facilitating the factorization of both forward and backward proof steps. The second system, called flower calculus, is an intuitionistic refinement of C. S. Peirce's theory of existential graphs. Both types of systems are shown to be analytic and fully invertible, making them amenable to proof automation techniques. We finally develop the Flower Prover, another prototype of GUI for interactive proof construction based on the flower calculus.

Institut Polytechnique de Paris
91120 Palaiseau, France