発表概要

証明支援系 Coq のプログラムに対する対話的修正機構の提案

森口草介[†1]　渡部卓雄[†1]

証明支援系 Coq のプログラムに対する対話的修正機構を提案する．証明支援系はプログラムなどの正しさを証明するために利用されている．しかし，Coq による証明は対象となる体系の定義に密接に関連しているため，定義のわずかな修正に対して多くの不整合が発生する．このような不整合に対し，場所の確認を Coq からのエラーによって発見するという，退屈で，かつ誤りを組み込みやすい作業で行うことが一般的になっている．提案する機構を組み込んだ証明支援系を用いることにより，この作業を簡易化することができる．対話的修正機構は修正箇所の記述を受け取ることで，その修正によって不整合が生じた箇所を提示することができる．また，この提示された箇所をもとに修正を対話的に行うことを可能にしている．本発表で提案している Coq における対話的修正機構では，帰納型に新たなコンストラクタを追加することと，そのコンストラクタに対応するデストラクタの拡張を行う機能を持っている．このように機能としては非常に制限されているが，それにより非常に高い再利用性を得ることができる．本発表では，不整合のある箇所の検出方法と，制限の詳細について述べる．

Towards on an Interactive Refinement Mechanism for Coq Scripts

Sosuke Moriguchi[†1] and Takuo Watanabe[†1]

We propose an interactive refinement mechanism for Coq proof assistant, which aims to improve the reusability of proof scripts. Suppose that we have some proof scripts about a computer program. We call the program the proof target of the proof scripts. The problem is that it is generally difficult to reuse the scripts for a modified proof target. Even some small changes, such as just adding new constructors to some inductive types, could cause a large number of inconsistencies in the scripts. Usually we should manually find and fix them by examining error messages issued by the proof assistant, which is hard and error-prone process. Using the proof assistant modified to incorporate the proposed mechanism can ease the process. By providing the modified part of the target definition, our interactive refinement mechanism finds all the inconsistent parts in the definition and the proof scripts. After finding them, we can easily gain the proofs for the new target by issuing correction commands to the modified proof assistant. The interactive refinement mechanism for Coq described in this presentation covers the extension of inductive data types with new constructors and the destructors correspond to them. Our current refinement mechanism is limited, but provides us with practical means to reuse machine-assisted proofs. In this presentation, we discuss the methods to detect inconsistent parts and details of limitations for extensions.

（平成 23 年 4 月 26 日発表）

†1 東京工業大学大学院情報理工学研究科計算工学専攻
Department of Computer Science, Graduate School of Information Science and Engineering, Tokyo Institute of Technology