情報処理学会論文誌　プログラミング　Vol. 1　No. 2　123 (Sep. 2008)

発表概要

再帰的定義を可能にする述語論理の証明支援系上の実装

矢田部　俊介[†1]

型付けを持つ証明支援系において，任意有限個の変数を持つ述語を含み，さらに量化子による変数の束縛が可能となる形式的体系（データ型やデータ構造）を定義することが困難であることはよく知られている．さらに，束縛された変数への具体的な値を代入する操作を証明支援系が解釈することにも困難がともなう．そのため，量化子を含む論理式によって関数を再帰的に定義しようとして，たとえそれが人間の目には十全な定義であっても，証明支援系はその関数の停止判定に失敗することが起こる．本発表では，Martin-Löf の直観主義的型理論に基づいた厳格な型付けを持つ証明支援系 Agda 上で以上の点を解決した述語論理の実装例を紹介する．この例においては，多変数述語の実装は，依存型理論（dependent type theory）を使用した．すなわち，自由変数の個数と，そのうちの束縛された変数の個数を，同時に型に情報として含むように，多変数述語のデータ型を定義した．また，束縛された変数の個数のうえでの帰納法により，知識様相述語論理の充足関係を関数として定義した．このことにより，関数定義が明示的に有限回で停止することを証明支援系に理解させることができ，関数の停止判定に成功した．

An Implementation of Predicate Logic in a Proof Assistant for Supporting Recursive Definitions

Shunsuke Yatabe[†1]

It is difficult to implement a data structure, as a predicate logic which has predicates with bound variables, to proof assistants with types. In particular, it is very difficult to implement the substitution of terms for bound variables to the proof assistant. Sometimes it fails to pass a termination check of a function which is recursively defined by a formula, containing such substitutions, which is well-defined to the human eye. In our presentation, we give an example of an implementation of predicate logic in Agda, a proof assistant which is based on a constructive type theory a la Martin-Löf. We used dependent types to define predicates. The type (of predicates) contains two kinds of information: How many variables the predicate has, and how many ones of them are bound. As for the termination check, we give an example of a definition of a satisfaction function for a knowledge modal logic system. Our inductive definition passed the termination check of Agda because we used an induction on a number of bound variables and it explicitly shows that the reduction ends in finite steps.

（平成 20 年 3 月 17 日発表）

†1 産業技術総合研究所システム検証研究センター
Research Center for Verification and Semantics (CVS), National Institute of Advanced Industrial Science and Technology (AIST)