Relating Computational Effects by ⊤⊤-Lifting

Shin-ya Katsumata

Research Institute for Mathematical Sciences
Kyoto University, Kyoto, 606-8502, Japan
sinya@kurims.kyoto-u.ac.jp

Abstract. We consider the problem of establishing a relationship between two monadic semantics of the λc-calculus extended with algebraic operations. We show that two monadic semantics of any program are related if the relation includes value relations and is closed under the algebraic operations in the λc-calculus.

1 Introduction

Suppose that two monadic semantics A1, A2 are given to a call-by-value functional language. We often encounter the problem of establishing a certain relationship between two monadic representations of computational effects caused by programs. In this paper, we name it effect simulation problem and formulate it as follows:

Let V be a relation between values in A1 and A2, and O be a relation between computational effects in A1 and A2. Then for any open term x1 : b1, ⋯ , xn : bn ⊢ M : b (where bi and b are all base types), do two denotations A1[M] and A2[M] map the values related by V to the computational effects related by O?

This problem has been considered in various forms. For instance:

– From a monadic semantics A of the language by a monad T, we can build another monadic semantics A_{CPS} by the continuation monad (→ TR) ⇒ TR. The latter semantics is called CPS semantics. In [1], Filinski formally showed monadic congruence result that establishes a relationship between monadic semantics and CPS semantics, which is summarised by the equation A_{CPS} [M] = λk . k^k(A[M]).
– The effect simulation problem is one of the main topics in the recent work [2] by Filinski. There, it is studied the relationships between two monadic semantics of a functional language with recursive types and computational effects.

We tackle the effect simulation problem under the situation where 1) the call-by-value functional language is the simply typed λc-calculus with products, coproducts and algebraic operations [6] and 2) the semantics is given in any bi-CCC with a strong monad and algebraic operations. We then show that the answer of the simulation problem is “yes” if the simulation O includes V (via monad units) and is closed under the algebraic operations in the λc-calculus. The point of this result is the generality; it holds with any monad, algebraic operation and relation V and O. The proof employs categorical ⊤⊤-lifting [3], which is a semantic analogue of the leapfrog method introduced by Lindley and Stark [4, 5], to construct a logical relation between two monadic semantics.
References