

発表概要

動的環境と限定継続を持つプログラム言語の意味論と実装

田中 陽[†] 亀山 幸義[†]

本研究は、関数型言語 Scheme における動的環境と限定継続の共存について検討し、形式的意味論を定義し、それに基づく実装を与えたものである。動的環境はプログラム実行時に動的に決定される環境で、Scheme では、プログラム中の手続きが一定の動的環境を持つことを保証するための機構として、dynamic-wind が用意されている。限定継続は、「計算の残りの一部」のことである。Scheme の標準手続き call/cc が「計算の残り」全体を操作するのに対して、本研究で扱う shift/reset はこの限定継続を操作し、種々の探索問題などがより簡潔に記述できるようになる。すでに知られているように、dynamic-wind と call/cc の共存は容易ではない。Scheme の仕様書 R5RS の形式的意味論はこれらの共存に対応しておらず、後の研究で修正された。我々は、shift/reset を Scheme に追加し、記述力を向上させる研究を行っている。本研究はその一環として、dynamic-wind と shift/reset の共存について取り組んだものである。まず、R5RS の表示の意味論を拡張して、shift/reset に対応した意味論を与える。次に、プログラムの実行が dynamic-wind の性質を保証することを示すために、その意味論に対応する抽象機械を導く。またあわせて、この意味論に基づいた Scheme インタプリタを作成し、shift/reset と dynamic-wind を含むプログラムが正しく動くことを確かめた。

Formal Semantics and Implementation of Dynamic Environments and Delimited Continuations

YO TANAKA[†] and YUKIYOSHI KAMEYAMA[†]

In this presentation, we address the problem of coexistence of dynamic environments and delimited continuations in the programming language Scheme, and in particular, we define a formal semantics and an implementation for a language extended with them. A dynamic environment is an environment which is determined dynamically, that is, at the execution time. Scheme provides the procedure dynamic-wind by which one can guarantee that a certain procedure always has a certain dynamic environment. The delimited continuation is (the specific) part of the rest of computation. While call/cc in Scheme provides an access to the continuation, the rest of computation, shift/reset, the subject of our study, access to the delimited continuation, hence, one can compactly represent various search algorithms in term of shift/reset. As is well known, having both dynamic-wind and call/cc in a single language is problematic. R5RS, the standard specification of Scheme, has a bug in its denotational semantics, and a later study has fixed this bug. Our long-standing aim is to add shift/reset (or equivalent) to various programming languages so that we can get more expressivity, and as a first step of this aim, we have developed a solution of this combination. We first extended the R5RS formal semantics to the one including shift/reset. Then, to prove that the execution guarantees the necessary properties for dynamic-wind, we derived an abstract machine which corresponds to our formal semantics. We also implemented a Scheme interpreter based on the semantics, and confirmed that programs which have both dynamic-wind and shift/reset ran as we expected.

(平成 19 年 3 月 22 日発表)

[†] 筑波大学大学院システム情報工学研究科
Graduate School of Systems and Information Engineering,
University of Tsukuba