Abstract

Delimited continuations are a programming language concept which represents the prefix of the evaluation context in a given state of the program execution. Control operators for delimited continuations enable advanced manipulation of the program’s runtime state, e.g., backtracking, coroutines. The most well-known control operators for delimited continuations are Felleisen’s control/prompt and Danvy and Filinski’s shift/reset. Delimited continuations found many applications, both theoretical and practical, including: mobile computing, operating systems, linguistics, partial evaluation, representing monads.

The subject of this thesis is a variant of the Danvy and Filinski’s shift/reset, which is called shift0/reset0 in the literature. They differ from shift/reset in that they give access to layers of continuations arranged as a stack. We present numerous results for these operators, including:

**Type system.** There exists an expressive type system for shift0/reset0, which allows to precisely characterize the control effects executed by a given expression; in particular it allows to distinguish pure expressions from impure ones which can have control effects. The type system has type preservation, progress and termination properties, it also conservatively extends the Danvy and Filinski’s type system for shift/reset.

**CPS translations.** There exist simple and elegant CPS translations for shift0/reset0 control operators: an untyped and a type-directed one. The second one is selective: it does not modify the expressions classified by the type system as pure. One can derive the classic CPS translation for shift/reset from the type-directed CPS translation.

**Equational axiomatization.** There exists a simple equational axiomatization of shift0/reset0 which is sound and complete with respect to the untyped CPS translation. The result is easily extended to the typed setting.

**Interpretation of the CPS hierarchy.** The control operators for the CPS hierarchy of Danvy and Filinski generalize the shift/reset operators. The CPS hierarchy can be expressed using shift0/reset0, both in untyped and typed settings.

**Implementation.** The shift0/reset0 control operators can be efficiently implemented on popular computer architectures. We present a proof-of-concept implementation on the x86-64 architecture.

The results listed above suggest that the shift0/reset0 have fundamental importance, and are therefore very important in the landscape of control operators for delimited continuations.