Continuations and Martin-Löf's Type Theory

A thesis presented in partial fulfilment of the requirements of the degree of

Doctor of Philosophy
in
Computer Science

at Massey University, Albany, New Zealand.

Neil Leslie
February 29, 2000

Abstract

We explain how to program with continuations in Martin-Löf's theory of types (M-LTT).

M-LTT is a theory designed to formalize constructive mathematics. By appealing to the Curry-Howard 'propositions as types' analogy, and to the Brouwer-Heyting-Kolmogorov interpretation of intuitionistic logic we can treat M-LTT as a framework for the specification and derivation of correct functional programs. However, programming in M-LTT has two weaknesses:

- we are limited in the functions that we can naturally express;
- the functions that we do write naturally are often inefficient.

Programming with continuations allows us partially to address these problems. The continuation-passing programming style is known to offer a number of advantages to the functional programmer. We can also observe a relationship between continuation passing and type lifting in categorial grammar.

We present computation rules which allow us to use continuations with inductively-defined types, and with types not presented inductively. We justify the new elimination rules using the usual proof-theoretic semantics. We show that the new rules preserve the consistency of the theory.

We show how to use well-orderings to encode continuation-passing operators for inductively defined types.

Acknowledgements

An earlier version of some of the material in Chapter 6 appeared as [70].

I would like to thank:

- Peter Kay of Massey University's Albany campus, and Steve Reeves of Waikato University, for providing invaluable support and guidance;
- Ross Renner, and the School of Mathematical and Computing Sciences at Victoria University of Wellington, for indulging me with time and money to visit Waikato University to talk with Steve;
- Mhairi for caring (again) that I should finish a thesis;
- Keir and Ailidh for not caring about theses at all.

I would also like to acknowledge the debt which I owe to Aaron T. Beck.

Contents

List of Rules

List of Programs

List of Figures