1 — Introduction

Explanation of problem space: need and motivation demonstrated with examples.

What are exceptions? How are they typed? What have approaches been before?

van Bakel and the λ^{try}-calculus is different approach. λ^{try} already compared to the 'classical- cal...

Exceptions have been done but unnamed or dispatch on type. λ^{try} introduces exceptions with names.

Features of computer programs are discovered twice: by logicians and by computer scientists.[**?**] Exceptions have been mapped to continuations which have been mapped to classical logic. What about a calculus that models exceptions directly? HOw does it behave?

1.1 Solution

Use van Bakel's translation of λ^{try} to $\lambda\mu$. Define a translation from $\lambda\mu$ to CDC, which closely models Haskell's syntax. Write a CDC interpreter for generating derivations that can be transcribed into proofs. Investigate properties of $\lambda\mu$ translation. Use this translation to find a translation from λ^{try} to lmu. Implement λ^{try} directly in Haskell by following this.

1.2 Contributions

This paper makes the following contributions:

- Haskell interpreter for a calculus of delimited continuations, CDC, written by SPJ
- Translation of $\lambda\mu$ to CDC along with proof of soundness and completeness with respect to mu reduction.
- A translation of λ^{try} to CDC.
- A proof of concept implementation of λ^{try} in Haskell, based on this translation.

- The specification for a language extension for named exceptions in Haskell, based on λ^{try}.

Bibliography

[1] Philippe de Groote. A cps-translation of the lambda-μ-calculus. In *Trees in Algebra and Programming - CAAP'94, 19th International Colloquium, Edinburgh, U.K., April 11-13, 1994, Proceedings*, pages 85–99, 1994.

[2] R. Kent Dybvig, Simon L. Peyton Jones, and Amr Sabry. A monadic framework for delimited continuations. *J. Funct. Program.*, 17(6):687–730, 2007.

[3] Jean-Yves Girard, Yves Lafont, and Paul Taylor. *Proofs and Types*. Cambridge University Press, 1989.

[4] Michel Parigot. Lambda-my-calculus: An algorithmic interpretation of classical natural deduction. In *Logic Programming and Automated Reasoning,International Conference LPAR'92, St. Petersburg, Russia, July 15-20, 1992, Proceedings*, pages 190–201, 1992.

[5] Steffen van Bakel. λ^{try}: exception handling with failure and recovery, 2015. Unpublished paper on formally modelling exception handling in the λ-calculus.

[6] Philip Wadler. Propositions as types. *Commun. ACM*, 58(12):75–84, 2015.