Parametricity for Haskell with Imprecise Error Semantics

Florian Stenger and Janis Voigtländer

Institut für Theoretische Informatik
Technische Universität Dresden
01062 Dresden, Germany

{stenger,voigt}@tcs.inf.tu-dresden.de

Abstract

Error raising, propagation, and handling in the functional programming language Haskell can be imprecise in the sense that a language implementation's choice of local evaluation order, and optimising transformations to apply, may influence which of a number of potential failure events hidden somewhere in a program is actually triggered. While this has pragmatic advantages from an implementation point of view, it also complicates the meaning of programs and thus requires extra care when reasoning about them. The proper semantic setup is one in which every erroneous value represents a whole set of potential (but not arbitrary) failure causes [2]. The associated propagation rules are somewhat askew to standard notions of program flow and value dependence. As a consequence, standard reasoning techniques are cast into doubt, and rightly so. We study this issue for one such reasoning technique, namely the derivation of (equational and inequational) free theorems from polymorphic types [3, 4]. Continuing earlier work [1], we revise and extend the foundational notion of relational parametricity, as well as further material required to make it applicable. More generally, we believe that our new development and proofs help direct the way for incorporating further and other extensions and semantic features that deviate from the "naive" setting in which reasoning about Haskell programs often takes place.

References

1. P. Johann and J. Voigtländer. Free theorems in the presence of seq. In *Principles of Programming Languages, Proceedings*, pages 99–110. ACM Press, 2004.
2. S.L. Peyton Jones, A. Reid, C.A.R. Hoare, S. Marlow, and F. Henderson. A semantics for imprecise exceptions. In *Programming Language Design and Implementation, Proceedings*, pages 25–36. ACM Press, 1999.
3. J.C. Reynolds. Types, abstraction and parametric polymorphism. In *Information Processing, Proceedings*, pages 513–523. Elsevier, 1983.
4. P. Wadler. Theorems for free! In *Functional Programming Languages and Computer Architecture, Proceedings*, pages 347–359. ACM Press, 1989.