

*Typing Haskell in Haskell**

MARK P. JONES

*Pacific Software Research Center
Department of Computer Science and Engineering
Oregon Graduate Institute of Science and Technology
20000 NW Walker Road, Beaverton, OR 97006, USA
(e-mail: mpj@cse.ogi.edu)*

Abstract

Haskell benefits from a sophisticated type system, but implementors, programmers, and researchers suffer because it has no formal description. To remedy this shortcoming, we present a Haskell program that implements a Haskell typechecker, thus providing a mathematically rigorous specification in a notation that is familiar to Haskell users. We expect this program to fill a serious gap in current descriptions of Haskell, both as a starting point for discussions about existing features of the type system, and as a platform from which to explore new proposals.

1 Introduction

Haskell¹ benefits from one of the most sophisticated type systems of any widely used programming language. Unfortunately, it also suffers because there is no formal specification of what the type system should be. As a result:

- It is hard for Haskell implementors to be sure that their systems accept the same programs as other implementations. The informal specification in the Haskell report (Peyton Jones & Hughes, 1999) leaves too much room for confusion and misinterpretation. This leads to genuine discrepancies between implementations, as subscribers to the Haskell mailing list will have seen.
- It is hard for Haskell programmers to understand the details of the type system and to appreciate why some programs are accepted when others are not. Formal presentations of most aspects of the type system are available, but they often abstract on specific features that are Haskell-like, but not Haskell-exact, and do not describe the complete type system. Moreover, these papers tend to use disparate and unfamiliar technical notation and concepts that may be difficult for some Haskell programmers to understand.

* An earlier version of this paper was presented at the Haskell workshop in Paris, France, on October 1, 2000. Both papers describe the same type system, but some significant parts of this version have been rewritten, restructured, or expanded to clarify the presentation.

¹ Throughout, we use ‘Haskell’ as an abbreviation for ‘Haskell 98’.