

Dependently Typed Languages in Statix

Jonathan Brouwer ✉🏠

Delft University of Technology, The Netherlands

Jesper Cockx ✉🏠

Delft University of Technology, The Netherlands

Aron Zwaan ✉🏠

Delft University of Technology, The Netherlands

Abstract

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Praesent convallis orci arcu, eu mollis dolor. Aliquam eleifend suscipit lacinia. Maecenas quam mi, porta ut lacinia sed, convallis ac dui. Lorem ipsum dolor sit amet, consectetur adipiscing elit. Suspendisse potenti.

2012 ACM Subject Classification Replace ccsdesc macro with valid one

Keywords and phrases Spoofox, Statix, Dependent Types

Digital Object Identifier 10.4230/OASICS.CVIT.2016.23

1 Introduction

Spoofox is a textual language workbench: a collection of tools that enable the development of textual languages. When working with the Spoofox workbench, the Statix meta-language can be used for the specification of Static Semantics. To provide these advantages to as many language developers as possible, Statix aims to cover a broad range of languages and type-systems. However, no attempts have been made to express dependently typed languages in Statix.

Dependently typed languages are different from other languages because they allow types to be parameterized by values. This allows more rigorous reasoning over types and the values that are inhabited by a type. This expressiveness also makes dependent type systems more complicated to implement. Especially, deciding equality of types requires evaluation of the terms they are parameterized by.

This goal of this paper is to investigate how well Statix is fit for the task of defining a simple dependently-typed language. We want to investigate whether typical features of dependently typed languages can be encoded concisely in Statix. The goal is not to show that Statix can implement it, but that implementing it is easier in Statix than in a general-purpose programming language.

We will first show the base language and explain the way that Statix was used to implement this language. Next, we will explore several features and see how well they can be expressed in Statix.

2 Background

References

- 1 Edsger W. Dijkstra. Letters to the editor: go to statement considered harmful. *Commun. ACM*, 11(3):147–148, 1968. doi:10.1145/362929.362947.
- 2 Jim Gray and Andreas Reuter. *Transaction Processing: Concepts and Techniques*. Morgan Kaufmann, 1993.
- 3 John E. Hopcroft, Wolfgang J. Paul, and Leslie G. Valiant. On time versus space and related problems. In *16th Annual Symposium on Foundations of Computer Science, Berkeley*,



© Jane Open Access and Joan R. Public;
licensed under Creative Commons License CC-BY 4.0

42nd Conference on Very Important Topics (CVIT 2016).

Editors: John Q. Open and Joan R. Access; Article No. 23; pp. 23:1–23:2

OpenAccess Series in Informatics



OASICS Schloss Dagstuhl – Leibniz-Zentrum für Informatik, Dagstuhl Publishing, Germany

California, USA, October 13-15, 1975, pages 57–64. IEEE Computer Society, 1975. doi:
10.1109/SFCS.1975.23.

- 4 Donald E. Knuth. Computer Programming as an Art. *Commun. ACM*, 17(12):667–673, 1974.
doi:10.1145/361604.361612.