My Report!

First year review report

Zhili Tian

Supervised by Prof. Thorsten Altenkich

& Prof. Ulrik Buchholtz

Funtional Programming Lab

School of Computer Science

University of Nottingham

May 26, 2025

Abstract

Giving a short overview of the work in your project.[1]

Contents

Chapter 1

Introduction

1.1 Background and Motivation

One fundamental feature of programming languages is their type system, which allows classification of values and prevents potential type errors. Concrete examples of types include *unit*, *boolean*, *natural number*, etc. Additionally, there is bigger class of types which enable the construction of higher-kinded types by parameterizing over other types, such as *list*, *binary tree*, *maybe*, etc.

Here is a piece of agda code:

`This is code`

```
data List (X : Set) : Set where
  [] : List X
  _::_ : X → List X → List X

_++_ : {X : Set} → List X → List X → List X
[] ++ ys = ys
(x :: xs) ++ ys = x :: (xs ++ ys)

data BTree (X : Set) : Set where
  leaf : BTree X
  node : BTree X → X → BTree X → BTree X

flatten : {X : Set} → BTree X → List X
flatten leaf = []
flatten (node lt x rt) = flatten lt ++ (x :: flatten rt)
```

1.2 Aims and Objectives

1.3 Overview of the Report

Chapter 2

Prerequisites

```
two : ℕ
two = suc (suc zero)

proof : zero ≡ zero
proof = refl
```

2.1 Type Theory

This is type theory

Chapter 3

Conducted Research

3.1 Literature review

3.2 Topics Studied

3.3 Questions

Chapter 4

Future Work Plan

This is future work plan.

Chapter 5

Conclusions

This is conclusions.

Chapter 6

Appendix

This is appendix.

Bibliography

[1] Abbott, M., Altenkirch, T., and Ghani, N. Containers: Constructing strictly positive types. *Theoretical Computer Science 342*, 1 (2005), 3–27. Applied Semantics: Selected Topics.