Formalizing operational semantics in a proof assistant

Edoardo Marangoni
University of Milan

A thesis submitted for the degree of
Master of Science

September 13, 2023

Contents

CHAPTER 1

Introduction

CHAPTER 2

Semantics

2.1 Introduction

In 1967, computer scientist Robert Floyd wrote, in a seminal paper [Flo67],

CHAPTER 3

Induction and coinduction

3.1 Introduction

3.2 Infinite datatypes

3.3 Infinite proofs

3.4 Relation with fixed points

CHAPTER 4

Agda

4.1 Dependent types

4.2 Termination and productivity

4.3 Sized types

CHAPTER 5

The partiality monad

5.1 Monads

5.2 Implementation

CHAPTER 6

The IMP language

In this chapter we will go over the implementation of a simple imperative language called **Imp**, as described in [PdAC$^+$23]. After defining its syntax, we will give rules for its semantics and show its implementation in Agda. After this introductory work, we will discuss analysis and optimization of Imp programs.

6.1 Syntax

The syntax of the Imp language is straightforward, and can be described in a handful of EBNF rules.

6.2 Semantics

6.3 Implementation

6.4 Analysis and optimization

6.4.1 Definite initialization analysis

6.4.2 Pure folding

Bibliography

[Flo67] R. W. Floyd. Assigning meanings to programs. *Mathematical aspects of computer science*, 19, 1967.

[PdAC+23] Benjamin C. Pierce, Arthur Azevedo de Amorim, Chris Casinghino, Marco Gaboardi, Michael Greenberg, Cătălin Hriţcu, Vilhelm Sjöberg, and Brent Yorgey. Logical foundations, 2023. [Accessed 13-09-2023].