Formalization of the Denotation Semantics of Automatic Differentiation in Coq Project Proposal

Curtis Chin Jen Sem
(5601118)

April 2020

1 Introduction

AI and machine learning research has sparked a lot of new interest in recent times due to its many applications and ability to solve very difficult problems. One of the principle techniques in practice is the use of gradient descent, which tries optimize some problem by trying to calculate the local minimum of a differentiable function.

This is regularly done using a technique called Automatic Differentiation. There has been a recent surge of interest in formulating languages for defining automatic differentiable functions. This could have many benefits such as both applying many of the established high and low level optimizations known in programming languages research, ease defining functions for use in a gradient descent optimization through higher order functions and correctness through the use of a possible type system.

We aim to formalize an extendable proof of automatic differentiation on a simply typed lambda calculus in the proof assistant Coq, opening up further possibilities for formally proving the correctness of more complex language features in the future.

2 Background

2.1 Automatic Differentiation

The purpose of automatic differentiation is to calculate the derivative of a function, or in other words, calculate the derivative of function described by a arbitrarily large program. So the semantics which one would normally expect in programming language is extended with relevant concepts such as derivative values and the chain rule.

Automatic or algorithmic differentiation is beneficial over other methods of automatically calculating the derivatives of functions such as numerical differentiation or symbolic differentiation due to its balance between speed and computational complexity.

There are two main modes of automatic differentiation which I will both discuss. These are namely forward or