CS51 Final Project

Evangelos Kassos

May 2022

1 Introduction

In this short paper, I describe the implementation of the extensions that can be found on my final project code. I added additional expression types, added additional unop and binop operation types, and implemented a lexically scoped environment semantics.

2 Additional expression types

Because the distribution code focused on integers and booleans, I thought that it would be great to incorporate more of the common types we saw this semester. I implemented the float type as an expression type, and further operations, as described below.

3 Additional unop and binop operation types

To complement the distributed unop and binop operation types, I added the following binop operation types: over (division), DNE (does not equal), less than, and their float counterparts. I also included the negate counterparts of booleans and floats for greater compatibility across data types. This addition should add to the usability of MiniML.

4 A lexically scoped environment semantics

Finally, and most importantly, I implemented a lexically scoped environment semantics, as described in the textbook. This is where I spent most of my time, trying to combine the functionality of dynamically and lexically scoped environment semantics. In the end, I combined most functions into a single call for both scopes, with a few functions, like let rec having different calls because of the different scoping mechanisms. Both scoped environment semantics support all operations and types described above.

5 Future directions

A project, especially a personal one, like this one, is never really over. I would like to add additional data types, including lists of all described data types.