

CS345H PA4 Project Proposal

Zack Misso, Nimish Kumar

November 9, 2015

1 Type Inference

We will implement basic type inference for all aspects of L except for lists. We are initially going to simplify lists to contain only *ints* or only *strings*. Constant types will be either *ints* or *strings*. *Nil* will be considered its own type, and will not be allowed as a member of a list.

2 Merge Interpreter and Type Inference

Before working on more improvements to L, we will have to merge all of our previous projects with PA4 so that we can change the lexer, parser, and interpreter as well.

3 Polymorphic Lists

We will make our type inference system support polymorphic lists, which will involve supporting lists that contain more than one type, while still having a sound type system.

4 Increase Performance and Efficiency

We will try to improve our understanding of our lexer, parser, and interpreter and locate areas where they can be improved in comparison to the ones provided by Dr. Dillig. For instance, there is probably a lot of overhead in Flex and Bison that we could remove. Performance can be tested by timing mergesort on a large list of ints.

5 Add/Change L Operations

We will change the *lambda* keyword to something shorter, such as a */*. We will also add a looping construct that will run a given expression a number of times, similar to a *for* loop. We will add functionality to allow for negative numbers in L programs.

6 User-Annotated Types

We want to allow users to define the types of input and output parameters for some functions, so that type inference can be faster.