# Homework 3 (Semantics)

*CS 381, Spring 2019*

June 16, 2019

Prepared by

Adam Stewart *(stewaada)*
Anish Asrani *(asrania)*
Lucas Frey *(freyl)*
Mazen Alotaibi *(alotaima)*
Sergei Poliakov *(poliakos)*

## CONTENTS

## 1 Exercise 1. A Stack Language

**1.1 Define the semantics for the stack language as a Haskell function sem that yields the semantics of a program. Please note that the semantic domain has to be defined as a function domain (since the meaning of a stack program is a transformation of stacks) and as an error domain (since operations can fail). Therefore, sem has to have the following type where you have to find an appropriate type defintition for D.**

## 2 Exercise 2. Extending the Stack Language by Macros

**2.1 Extend the abstract syntax to represent macro definitions and calls, that is, give a correspondingly changed data definition for Cmd.**

**2.2 Define a new type State to represent the state for the new language. The state includes the macro definitions and the stack. Please note that a macro definition can be represented by a pair whose first component is the macro name and the second component is the sequence of commands. Multiple macro definitions can be stored in a list. A type to represent macro definitions could thus be defined as follows.**

**2.3 Define the semantics for the extended language as a function sem2. As in exercise 1, you probably want to define an auxiliary function semCmd2 for the semantics of individual operations.**

## 3 Exercise 3. Designing Abstract Syntax

**3.1 Define the semantics of Mini Logo by giving two function definitions. First, define a function semS that has the following type**