Ioannis Nearchou

Jonah Kubath

CS 5800 Theory Foundations

11/4/2018

**CS 5800 Project Preliminary Design: YACC/BISON Calculator Implementation**

This paper outlines the design of a proposed project to implement a yacc/bison description of a calculator.

**Project Approach**

As discussed in the previous project proposal, a calculator implementation using FLEX is designed. To implement the calculator, a grammar file will be written by this project group to handle expressions input by the user into the console. When an expression is being parsed by FLEX, the rules and transitions for parsed parts of the expression will be engaged, culminating in the return of a number to the console that is the calculation of the expression input by the user. Our possible approaches include writing a grammar that works for infix, prefix and postfix user expressions, for arithmetic and geometric mathematical expressions. The arithmetic and geometric expressions include standard multiplication, division, subtraction, addition, modulo, power, sine, cosine, tangent, etc. Possible modifications for basic calculus expressions and rules may be written later.

**Current Research Efforts**

Resources have been found for explaining parser grammar rules, including a class handout for a simple calculator and through <https://www.ibm.com/developerworks/aix/tutorials/au-lexyacc/index.html>, for FLEX in particular. These provide enough examples in what to consider for handling order of precedence with expressions, to govern what grammar we may write for our implementation.

**Design Flow**

The design of the project is as follows:

1. Download FLEX and associated tools.
2. Create calculator grammar and input parser files.
3. Test project by the three approaches to expression format, with generated input.