CSC 135 Computing Theory and Programming Languages

PL Assignment 4, Web Programming Recursive Descent Parser

Anthony Chavez

Introduction

The purpose of this assignment was to implement a recursive-descent recognizer with a web interface for the Backus-Naur form (BNF) grammar given below. We already wrote the pseudocode in PL Assignment 1, so this was a good opportunity to implement this pseudocode and develop some web programming skills along the way.

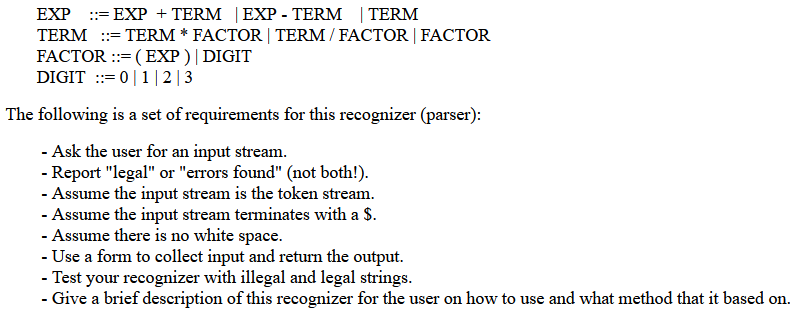


Figure 1: Given BNF Grammar and Parser requirements

Language Used and Experience

For my Web-based Recursive Descent Recognizer I initially attempted to write the parser in PHP, but I decided to use JavaScript since I am mainly familiar with Java. Although I have never used JavaScript, the two languages felt similar, and I only needed to understand some syntax differences for this project.

Overall, I spent two days learning PHP and implementing my parser with the language, but I did not have much success in getting the HTML file to run properly. Then I spent about a day learning JavaScript, implementing my parser, and debugging any runtime errors. In addition, JavaScript allowed easy debugging since I was able to convert the script to Java and use an IDE to find the few runtime errors in my parser logic.

Helpful Sources

I primarily used <https://www.w3schools.com/js/DEFAULT.asp> for all my JavaScript learning. The site does a great job of giving interactive examples that allow you to learn by doing instead of simply reading documentation.

Recommendations for Future Students

For future students, I recommend using JavaScript since I could not get PHP to work with my HTML file and JavaScript is much easier to understand. I also highly recommend coding the parser logic in Java after writing your pseudocode, so you can ensure your pseudocode logic is correct.