COMP 506, Spring 2019

Project 1 Questionnaire

Name: Nai-Fan Chen

NetId: nc41

1. Briefly discuss your experience building the scanner and parser. What things were easy to do? What things were hard?

Writing tokens is the easiest part because I just need to find the symbols in the Demo pdf and define them in the parser.y. scanner.l is simpler than parser.y since we just return the token which is identified by regular expression.

Error recovery and defining the syntax in parser are hard since I need to deal with ambiguity like if else, and the logic of error tokens is hard. Although error recovery is not that hard to implement, how to detect what error it made is difficult because there are lots of different situation. When I covered one situation, the other situation might be broken by certain reasons. Deciding to use yyerrok or yyclearin is not simple. If I chose wrong one, it might cause infinite loops.

1. Are there topics that we should cover in class to better prepare you for this project?

In the real world, compilers usually can show problems by very clear error message (not only the line number but variable name with detail explanation.). If I could get more detail about how flex and bison work, I might be able to show more variable error message which can really help users find and deal with the problem instantly.

Moreover, the shift/reduce conflict is the other topic I would love to know more. Now, I just type in the syntax that teacher gave in demo file. However, when I would like to add more feature to detect errors, it would cause conflict very often and I cannot fix it. Therefore, if I can get more relative knowledge in course, I probably can understand more how to design a context-free grammar on our own.

1. What, if anything, would you do differently if you could start over on this project?

I would not assign the symbols like ‘;’, ‘,’, ‘{’, etc as a token but just return the char back. I spend lots of time to define every symbol such as SEMI\_COLON, COLON and so on in order to return the recognized token which parser can understand. However, I just need to return that kind of symbol directly and the parser can recognize them as well. Hence, I will not need to define lots of tokens in the parser.y.