CS3320 – Mini Assignment 2

Name: Taha Adeel Mohammed
Roll Number: CS20BTECH11052

Q1) Write a short note on the following:

(i) 10 features that you notice in the Lex specification of C language.

1. Lex is a tool for analyzing the source code and converting the strings into tokens that can be parsed later on. Lex creates the tokens by matching the strings to regular expressions and returning the appropriate token.
2. In lex, the variables yyout and yyin are the output and input files respectively. The variable yytext is a char pointer that points to the source code/text to be analyzed.
3. The Lex specification is divided into three sections: definitions, rules and subroutines. These sections are separated by %%.
4. In the definition section, the digits(D), letters(L), hexadecimal digits(H), exponential numbers(E), and other substitutions are defined using regular expressions. When referencing to these substitutions, they are enclosed in curly braces[Eg: {L}({D}|{L})*].
5. C code in the definition section is written in between “%{“ and “%}” and it is copied directly to the top of the output file.
6. In the rules section, the patterns are declared to match the semantics of the C language code. These patterns match to keywords, variables (identifiers), literals, operators, whitespace, etc.
7. When the pattern is matched in the rules section, the C code inside the curly braces is executed. In the specification, we count() the matching pattern length and return the appropriate token.
8. In the subroutine section, the user defined functions are defined. yywrap() is a default function that is called when there is no more input to read.
9. The count() function counts the character length of the matching pattern taking into account whitespaces, and updates the column variable accordingly.
10. The comment() function scans the comment until it encounters “*/” and then exits, thereby functionally ignoring the comment.

(ii) 10 features that you notice in the Yacc specification of C language.

1. Yacc is a tool that helps parse a token stream/context free language into a syntax tree using grammar rules specified for the language.
2. Similarly to Lex, the input to yacc is divided into three sections: definitions, rules and subroutines. These sections are separated by %%.
3. In the definition section, the tokens are declared by using the keyword %token. These tokens are what are returned in the rules section in the Lex specification.
4. These tokens represent identifiers, constants, etc; operators such as incrementing, decrementing, left shift, etc; assigning operators(+=, -=, *=, &=, ^=, etc); storage class keywords(extern, static, etc); typenames(int, float, char, etc); jump keywords(if else, while, switch case, return, etc); etc.
5. In the rules section, the grammar for the language is defined. As the C language has a lot of features, the grammar to describe it completely is very elaborate and lengthy.
6. The expressions on the LHS are defined(in a recursive manner at times) by the expression after the colon. For example, expr: constant | expr ‘+’ expr implies expr is a constant or sum of exprs.
7. To be able to describe the various possible expressions using different operators, the specification has multiple grammar rules, defining primary_ expression, postfix_expression, unary_expression, unary_operators, multiplicative_expression, additive_expression, relational_expression, assignment_expression, assignment_operator, and many more such terms.
8. The grammar also has rules defining the storage_class_specifiers, type_specifiers, declaration_specifiers, specifiers for structs and unions, and various specifiers to define and initialize them.
9. There are also rules defining statements, hence letting the C language have features such as selection statements(if else, switch case, etc), iteration statements (while loop, for loop, etc), jump statements(break, return, etc) , functions and so on.
10. In the subroutine section, the functions used are defined here. The variable yytext[] is a char pointer pointing to the input text. The function yyerror(s) function prints the error message along with the column number onto the standard output.