BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI
DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Compiler Construction (CS F363)
II Semester 2023-24
Compiler Project
Coding Details
(March 5, 2024)

Group Number
28

1. Team Members Names and IDs

ID : **2020B3A70816P** Name : **Anishka Singh**

ID : **2020B3A71638P** Name : **Gautam Jajoo**

ID : **2020B3A71959P** Name : **Suraj Phalod**

ID : **2020B4A71986P** Name : **Ria Shekhawat**

ID : **2020B4A70830P** Name : **Karan Agrawal**

2. Mention the names of the Submitted files:

1. main.c
2. whole_include.h
3. vectors.h
4. vectors.h
5. vectorDef.h
6. vectors.c
7. utils.h
8. utils.c
9. printers.h
10. printers.c
11. strings.h
12. stringDef.h
13. strings.c
14. lexical_token.h
15. lexical_tokenDef.h
16. lexical_token.c
17. lexer.h
18. lexerDef.h
19. lexer.c
20. symbol_table.h
21. symbol_tableDef.h
22. symbol_table.c
23. rules.h
24. ruleDef.h
25. rules.c
26. parser.h
27. parserDef.h
28. parser.c
29. tree.h
30. treeDef.h
31. tree.c
32. driver.h
33. driver.c
34. grammar.txt
35. coding details.pdf
36. makefile
37. testcase1.txt
38. testcase2.txt
39. testcase3.txt
40. testcase4.txt
41. testcase5.txt
42. README.md

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : **42** (All files should be in ONE folder named as Group_#)

4. Have you compressed the folder as specified in the submission guidelines? (yes/no) : **YES**

5. **Lexer Details:**

[A]. Technique used for pattern matching: **We have created multiple functions corresponding to the different DFA states. We perform a retraction by manipulating the string and forward pointer wherever needed. Upon reaching the final state with respect to the function, the correct token is returned.**

[B]. Keyword Handling Technique: **We have a list of keywords**. **We compare each string value with each keyword value in the list.**

[C]. Hash function description, if used for keyword handling: **N/A, we have used a list of keywords.**

[D]. Have you used twin buffer? (yes/ no) : **YES**

[E]. Error handling and reporting (yes/No): **YES**

[F]. Describe the errors handled by you : **We have handled both lexical and syntax errors. Specific to Lexer, if a lexeme/input that is read doesn't correspond to any of the acceptable token types (essentially, not reaching the final state of the corresponding function, then we report it as error**).

[G]. Data Structure Description for tokenInfo (in maximum two lines):
A new struct named token has been created, with the attributes - int type, String lexeme_str, size_t line_num, size_t char_num, void *lexeme_value, char *error_msg. (here, String is another struct defined in the code)

6. **Parser Details:**

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):

i. grammar :**In the instance of the struct parser being used, its attribute 'grammar' is initialized as an instance of the struct 'grammar', which contains the following attributes with its corresponding type char *filename, Vector rules, Vector first, Vector follow, int *nullable, Vector parseTable;**

ii. FIRST and FOLLOW sets : **The struct parser has an attribute of thestruct grammar which further has attributes 'first' and 'follow'. Both of these are instances of the struct vector which stores elements of data type 'vector'. Each of these vectors consist of elements of the type struct 'token'**

iii. parse table:**The struct parser has an attribute of thestruct grammar which further has an attribute 'parseTable' which is an instances of the struct 'vector'**

iv. parse tree: (Describe the node structure also) : **struct treeNode - Token value; Vector children; TreeNode parent;**

v. Any other (specify and describe) : **A hash table with a corresponding hash function has been used to form a symbol table.**

[B]. Parse tree

i. Constructed (yes/no): **YES**

ii. Printing as per the given format (yes/no): **YES**

iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines)
IN-ORDER Traversal (leftmost subtree -> root -> rest of the subtrees)

[C]. Grammar and Computation of First and Follow Sets

i. Data structure for original grammar rules:**A struct 'vector' with attribute DATATYPE as RULE which corresponds to the struct 'rule'.**

ii. FIRST and FOLLOW sets computation automated (yes /no) : **YES**

iii. Name the functions (if automated) for computation of First and Follow sets: **populateFirst(), populateFollow()**

iv. If computed First and Follow sets manually and represented in file/function (name that) **NA**

[D]. Error Handling

v. Attempted (yes/ no):**YES**

vi. Describe the types of errors handled : **Lexical error, Syntactical error**

7. Compilation Details:

[A]. Makefile works (yes/no): **YES**

[B]. Code Compiles (yes/ no): **YES**

[C]. Mention the .c files that do not compile: **NA**

[D]. Any specific function that does not compile: **NA**

[E]. Ensured the compatibility of your code with the specified gcc version (yes/no) : **YES**

8. Driver Details: Does it take care of the options specified earlier(yes/no):**YES**
   [A]. status (describe in maximum 2 lines):**All the options in the driver function are working properly.**
   [B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: **N/A**

9. Specify the language features your lexer or parser is not able to handle (in maximum one line) **If the testcase begins with anything other than main or any user defined function, the parser does not parse the code.**

10. Are you availing the lifeline (Yes/No): **NO**

11. Declaration: We, **Anishka, Gautam, Suraj, Ria, Karan (Group 28)** declare that we have put our genuine efforts in creating the compiler project code and have submitted the code developed only by us. We have not copied any piece of code from any source. If our code is found plagiarized in any form or degree, we understand that a disciplinary action as per the institute rules will be taken against all of us in our team and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

Your names and IDs

ID : **2020B3A70816P** Name : **Anishka Singh**

ID : **2020B3A71638P** Name : **Gautam Jajoo**

ID : **2020B3A71959P** Name : **Suraj Phalod**

ID : **2020B4A71986P** Name : **Ria Shekhawat**

ID : **2020B4A70830P** Name : **Karan Agrawal**

Date: **05/03/2024**

Not to exceed 3 pages.