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: 10

1. Team Members Names and IDs

ID:	2021A7PS1463P	Name:	Dhyey Italiya
ID:	2021A7PS2434P	Name:	LAKSHIT SETHI
ID:	2021A7PS0523P	Name:	Abir Abhyankar
ID:	2021A7PS2414P	Name:	Saksham Verma
ID:	2021A7PS2412P	Name:	Saurabh Bhandari

2. Mention the names of the Submitted files :

1)driver.c
2)lexer.c
3)parser.c
4)lexer.h
5)parser.h
6)parserDef.h
7)lexerDef.h
8)makefile
9)trie.h
10)trie.c
11)vector.h
12)vector.c
13)utils.h
14)utils.c
15)coding details.pdf
16)grammar.txt
17)testcase1.txt
18)testcase2.txt
19)testcase3.txt
20)testcase4.txt
21)testcase5.txt
22)testcase6.txt
23)testcase7.txt
24)testcase8.txt
25)testcase9.txt

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : 25 (All files should be in ONE folder named as Group_#)

4. Have you compressed the folder as specified in the submission guidelines? YES

5. **Lexer Details:**

[A]. Technique used for pattern matching: DFA States and transitions are used for Pattern Matching with explicit checking for the length of the identifiers.

[B]. Keyword Handling Technique: Trie Data Structure

[C]. Hash function description, if used for keyword handling: We are using Trie instead of hashmap for optimal keyword handling, and also avoiding collisions which may occur with hashMap

[D]. Have you used twin buffers? : YES

[E]. Error handling and reporting : YES

[F]. Describe the errors handled by you: Unknown Pattern Error, Unknown Symbol Error, Variable Identifier Length Error and Function Identifier Length Error

[G]. Data Structure Description for tokenInfo (in maximum two lines): tokenInfo is a pointer to the struct TokenInfo, which maintains a list of tokens read from the TwinBuffer, and also maintains a count for the number of tokens read.

6. **Parser Details:**

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

i. grammar : Array of Linked List of Arrays. First dimension is the start variable, i.e. LHS(non-terminals) of grammar derivation . Second dimension is the Linked List of derivations and the third dimension is the corresponding derivation.

ii. FIRST and FOLLOW sets : 2 - Dimensional Array. First Dimension is Non- terminal and the second dimension is the corresponding terminals.

iii. parse table : 2 - Dimensional Array. First Dimension is Non- terminal and the second dimension is the corresponding terminals.

iv. parse tree: (Describe the node structure also) : It is a n-D tree. Each node contains details of the variable as well as the count of the number of children nodes and pointers to them.

v. Any other (specify and describe) : NO

[B]. Parse tree

i. Constructed : YES

ii. Printing as per the given format: YES

iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines) As instructed, Inorder (First we print the leftmost node and then we print the current node and then the rest of the children.)

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

i. Data structure for original grammar rules:Array of Linked List of Arrays.

ii. FIRST and FOLLOW sets computation automated : YES

iii. Name the functions (if automated) for computation of First and Follow sets: ComputeFirstAndFollowSets (As instructed)

iv. If computed First and Follow sets manually and represented in file/function (name that): None

[D]. Error Handling

v. Attempted: YES

vi. Describe the types of errors handled: Errors handled include errors like Misspelled Keyword, Missing Operator, Error in a defined structure, Unbalanced Parenthesis, etc. Parser is not falling into loop traps.

7. Compilation Details:

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

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

[C]. Mention the .c files that do not compile: None

[D]. Any specific function that does not compile: None

[E]. Ensured the compatibility of your code with the specified gcc version: YES

8. Driver Details: Does it take care of the options specified earlier: YES

9. Execution

[A]. status (described in maximum 2 lines): The code is compiling successfully and is working fully. All the modules and errors are handled perfectly. Error recovery is handled and heuristics are implemented successfully.

[B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the test case file name: NO

10. Specify the language features your lexer or parser is not able to handle (in maximum one line): None

11. Are you availing the lifeline: -

12. Declaration: We, Dhyey Italiya, Abir Abhyankar, Lakshit Sethi, Saksham Verma and Saurabh Bhandari 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:	2021A7PS1463P	Name:	Dhyey Italiya
ID:	2021A7PS2434P	Name:	LAKSHIT SETHI
ID:	2021A7PS0523P	Name:	Abir Abhyankar
ID:	2021A7PS2414P	Name:	Saksham Verma
ID:	2021A7PS2412P	Name:	Saurabh Bhandari

Date: **March 5, 2024.**

Not to exceed 3 pages.