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

Compiler Construction (CS F363)
II Semester 2021-22
Compiler Project (Stage-1 Submission)
Coding Details
(March 3, 2022)

Group Number
21

---

1. Team Members Names and IDs

ID: 2019A7PS0087P	**Name: Raghava Kasyap Kristipati**
ID: 2019A7PS0030P	**Name: K V S Preetam**
ID: 2019A7PS0068P	**Name: Yadagiri Shiva Sai Sashank**
ID: 2019A7PS0028P	**Name: Shanmukh Chandra Yama**
ID: 2019A7PS0083P	**Name: Uday Dheeraj Nulu**

2. Mention the names of the Submitted files :

1. lexer.c	7. Grammar.txt	13. t4.txt
2. lexer.h	8. coding details stage1.pdf	14. t5.txt
3. lexerDef.h	9. driver.c	15. t6.txt
4. parser.c	10. t1.txt	16. testcase1.txt
5. parser.h	11. t2.txt	17. testcase2.txt
6. parserDef.h	12. t3.txt	18. makefile

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : **18** (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: compared character by character
[B]. Keyword Handling Technique: Keywords are hashed and inserted into the Symbol Table at the start of the lexer and maximal munch technique is used for their handling.
[C]. Hash function description, if used for keyword handling: **sdbm** hash function has been used
[D]. Have you used twin buffer? (yes/ no) **YES**
[E]. Error handling and reporting (yes/No): **YES**
[F]. Describe the errors handled by you: Detected unknown letter if at all any got introduced into the code.
[G]. Data Structure Description for tokenInfo (in maximum two lines): structure was used for token info and the attributes are **line number**, **lexeme**, **value**, **isError**

6. Parser Details:

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

i. grammar: An array of array of linked list, **grammar[nonTerminalId][productionNumber] -> rule** gives a linked list which is the linked list of the grammar rule of nonTerminalId's production rule number.
ii. FIRST and FOLLOW sets: 2D array of bool type (using bool.h header file), where rows is the number of non-terminals and columns is number of terminals. **firstSet[i][j] = true** represents terminal 'j' exists in first set non-terminal 'i'. similar for follow set.
iii. parse table: 2D array of integers where row is the nonterminal id and column represents terminal id and **parseTable[i][j] = k**, represents k is the production no of 'i'th non terminal which has to matched if terminal 'j' is the input token.

iv. parse tree: (Describe the node structure also): **treeNode** is structure that contains information about the token, like lexeme, value etc. and contains array of pointers to each child. Tree is a structure which contains tree -> root which points to the root of the tree.

v. Any other (specify and describe): We have created a data structure which holds information about a non terminals appearance in a production rule which helped in calculations of follow sets. For example **rhsNonTerminalAppreance[nonTerminalId] = [[0, 1], [0,3]]**. Represents that nonterminalId appears on the right hand side of the production rules 1 and 3 of non-terminal with id '0'.

[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) Inorder traversal for nary tree has been implemented while printing all the children expected the last child are printed before the current node and then the last child is printed.

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

i. Data structure for original grammar rules : array of array of linked lists
ii. FIRST and FOLLOW sets computation automated (yes /no) **YES**
iii. Name the functions (if automated): for computation of First and Follow sets computeFirstAndFollowSets, computeFollowSetNonterminal, computeFollowSetRule, createNewBaseFollowSet, computeFirstSetTerminal, computeFirstSetNonTerminal, computeFirstSetRule.
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 errors: As specified in language specification are handled like length of variable name, fieldname etc. Panic mode is also implemented for error recovery and terminal terminal clash has been resolved by considering that terminal on top of stack has appeared instead of wrong input terminal.

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**
9. Execution
   [A]. status (describe in maximum 2 lines): all testcases are parsed successfully. Syntactic and lexical errors in t6 have also been reported correctly.
   [B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: no segmentation error for any test case. : **NA**

10. Specify the language features your lexer or parser is not able to handle (in maximum one line) **NO**

11. Are you availing the lifeline (Yes/No): **YES**

12. Declaration: We, **Raghava Kasyap Kristipati, Shanmukh Chandra Yama, Yadagiri Shiva Sai Sashank, K V S Preetam, Uday Dheeraj Nulu** 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: 2019A7PS0087P **Name: Raghava Kasyap Kristipati**
ID: 2019A7PS0030P **Name: K V S Preetam**
ID: 2019A7PS0068P **Name: Yadagiri Shiva Sai Sashank**
ID: 2019A7PS0028P **Name: Shanmukh Chandra Yama**
ID: 2019A7PS0083P **Name: Uday Dheeraj Nulu**

Date: **4/3/2022**

Not to exceed 3 pages.