**BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI**

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Compiler Construction (CS F363)**

**II Semester 2022-23**

**Compiler Project (Stage-1 Submission)**

**Coding Details** 

**(March 2, 2023)**

1. IDs and Names of team members

ID:\_\_\_\_2020A7PS0025P\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Shaz Furniturewala\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_2020A7PS0067P\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Niveditha Kovilath\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS0111P\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_Anshika Gupta\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS0112P\_\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_Sarthak Agarwal\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS1684P\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_Shreyas Shesham\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Mention the names of the Submitted files :

1\_\_\_\_lexer.c\_\_\_\_\_\_\_\_ 7\_\_\_\_ll1\_gram.c\_\_\_\_\_ 13\_\_parserDef.h\_\_\_ 19 \_\_\_\_adtDef.h\_\_\_\_

2\_\_\_\_lexer.h\_\_\_\_\_\_\_\_ 8\_\_\_\_ll1\_gram.h\_\_\_\_\_\_\_ 14\_\_grammarHash.h\_\_ 20\_\_\_\_Makefile\_\_\_\_\_

3\_\_\_\_parser.h\_\_\_\_\_\_\_\_ 9\_\_\_\_ll1\_gramDef.h\_\_\_\_\_\_\_ 15\_\_grammarHashDef.h\_\_

4\_\_\_\_parser.c\_\_\_\_\_\_\_\_ 10\_\_\_keywordTable.c\_\_\_\_ 16\_\_\_driver.c\_\_\_\_\_\_

5\_\_\_\_grammarHash.c\_ 11\_\_\_keywordTable.h\_\_\_\_ 17\_\_\_\_adt.c\_\_\_\_\_\_

6\_\_\_\_lexerDef.h\_\_\_\_\_ 12\_\_\_keywordTableDef.h\_\_\_\_ 18\_\_\_\_adt.h\_\_\_\_\_\_\_

1. Total number of submitted files: \_\_\_\_\_\_\_\_\_\_\_ (All files should be in **ONE folder** named exactly as Group\_#, # is your group number)
2. Have you mentioned your names and IDs at the top of each file (and commented well)? (Yes/ no) \_\_\_Yes\_\_ [Note: Files without names will not be evaluated]
3. Have you compressed the folder as specified in the submission guidelines? (yes/no)\_\_\_Yes\_\_\_\_\_\_
4. **Lexer Details:**
   1. Technique used for pattern matching: \_Pattern Matching is done using a switch case statement to check the appropriate lexeme for the next token\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. DFA implementation (State transition using switch case, graph, transition table, any other (specify):\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_State transition using switch case\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  2. Keyword Handling Technique: \_\_\_\_\_\_\_\_\_Hash Table search\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  3. Hash function description, if used for keyword handling:\_\_\_\_\_Rolling polynomial Hash Function\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  4. Have you used twin buffer? (yes/ no) \_\_\_\_\_\_\_\_\_\_\_\_\_Yes\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  5. Lexical error handling and reporting (yes/No):\_\_\_\_\_\_\_\_\_\_\_Yes\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
  6. Describe the lexical errors handled by you \_\_\_Invalid Character after various tokens, Invalid characters, Comment Mark not closed\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Data Structure Description for tokenInfo (in maximum two lines):

\_\_\_\_\_\_\_\_\_\_\_\_\_linked list\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Interface with parser \_\_\_\_\_\_\_Done, the token linked list is a global variable that is used in teh parser\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Parser Details:** 
   1. **High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):**
      1. grammar : \_\_\_\_Array of custom struct “lhs”, size = number of rules. “lhs” contains the id of the non-terminal present on the lhs of each rule, a pointer to the first symbol on the rhs (custom struct) as well as the last symbol on the rhs.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      2. parse table \_\_\_Array of ints of dimension Num-of-nonTerminals x Num-of-Terminals with each entry being a rule number.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      3. parse tree: (Describe the node structure also) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_The parse tree contains a single tree node root. Each tree-node has a union that becomes equal to either a terminal or non-terminal. It has a bool to define which one the union stores. It also points to its parent, it’s left child its right sibling and stores if it has been visited.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      4. Parsing Stack node structure :\_\_\_\_Each stack node contains a union of either a terminal or non-terminal and a variable defining which one of these is stored. It also points to the next stack node and its own location in the parse tree.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
      5. Any other (specify and describe) \_the rhs struct contains the id of the rhs, along with a bool to determine if its a terminal or non-terminal. It also contains a pointer to the next rhs and the previous rhs.
   2. **Parse tree** 
      1. Constructed (yes/no):\_\_Yes\_\_
      2. Printing as per the given format (yes/no): \_Yes\_
      3. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines)

\_\_\_\_\_\_\_\_Inorder traversal\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. **Grammar and Computation of First and Follow Sets** 
     1. Data structure for original grammar rules \_\_\_\_Array of custom struct lhs, which acts as the head of a double linked list of custom structs rhs.\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
     2. FIRST and FOLLOW sets computation automated (yes /no)\_\_\_\_\_\_\_Yes\_\_\_\_\_\_\_\_\_\_
     3. Data structure for representing sets \_\_\_\_\_\_\_\_\_Array\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
     4. Time complexity of computing FIRST sets \_\_\_\_O(NUM\_of\_RULES^2\*NUM\_of\_Terminals)
     5. Name the functions (if automated) for computation of First and Follow sets\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_computeFirstandFollow()\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
     6. If computed First and Follow sets manually and represented in file/function (name that) \_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_loadFirstAndFollow()\_\_\_\_\_\_\_\_\_\_
  2. **Error Handling** 
     1. Attempted (yes/ no):\_\_\_Yes\_\_\_
     2. Printing errors (All errors/ one at a time) :\_\_\_\_\_\_\_Yes\_\_\_\_\_\_
     3. Describe the types of errors handled

\_\_\_Input remaining, stack empty; Terminal mismatch; non-terminal doesnt exist; Linked empty, Stack not empty\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* + 1. Synchronizing tokens for error recovery (describe): We have implemented a retraction function after developing a synchronization set for all non-terminals
    2. Total number of errors detected in the given testcase t6(with\_syntax\_errors).txt \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_10\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. **Compilation Details:**
   1. Makefile works (yes/no):\_\_Yes\_\_\_
   2. Code Compiles (yes/ no):\_\_Yes\_\_\_\_
   3. Mention the .c files that do not compile: N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   4. Any specific function that does not compile: N/A\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
   5. Ensured the compatibility of your code with the specified gcc version(yes/no)\_\_\_Yes\_
2. **Driver Details**: Does it take care of the options specified earlier(yes/no):\_\_Yes\_\_\_
3. **Execution** 
   1. status (describe in maximum 2 lines):\_\_\_\_\_\_\_\_Running\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

* 1. Execution time taken for
     + t1.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_0.016s\_\_\_\_\_\_\_\_
     + t2.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_\_0.033s\_\_\_\_\_\_
     + t3.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_\_0.058s\_\_\_\_\_\_
     + t4.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_\_\_0.043s\_\_\_\_\_\_
     + t5.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_\_\_\_0.111s\_\_\_\_\_\_
     + t6.txt (in ticks) \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ and (in seconds) \_\_\_0.074s\_\_\_\_\_\_
  2. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_

1. Specify the language features your lexer or parser is not able to handle (in maximum one line)\_\_All specifications are being handled, \_\_\_\_\_\_\_\_ \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_
2. Are you availing the lifeline (Yes/No): \_\_\_Yes\_\_\_\_
3. Declaration: We, \_\_\_\_\_Shaz Furniturewala, Nivethida Kovilath, Anshika Gupta, Sarthak Agarwal and Shreyas Shesham\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ (your names), declare that we have put our genuine efforts into creating the compiler project code and have submitted the code developed only by our group. 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 disciplinary action as per the institute rules will be taken against us and we will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani. [Write your ID and name below]

ID:\_\_\_\_2020A7PS0025P\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Shaz Furniturewala\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_2020A7PS0067P\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_Niveditha Kovilath\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS0111P\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_Anshika Gupta\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS0112P\_\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_Sarthak Agarwal\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

ID:\_\_\_\_\_\_2020A7PS1684P\_\_\_\_\_\_\_\_\_\_Name:\_\_\_\_\_\_\_\_\_\_\_\_\_Shreyas Shesham\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_

Date: \_13/03/2023\_

---------------------------------------------------------------------------------------------------------------------------------------------

Should not exceed 4 pages.