

**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, 2022)**

**Group Number**

**25**

**1. Team Members Names and IDs**

ID __2021A7PS0576P__	Name__	Nek Manchanda__
ID __2021A7PS2429P__	Name__	Utkarsh Sharma__
ID __2021A7PS2683P__	Name__	Yash Sejpal__
ID __2021A7PS2689P__	Name__	Nishant Singh__
ID __2021A7PS2694P__	Name__	Prakhar Mundra__

**2. Mention the names of the Submitted files :**

1 driver.c	7 parser.h	13 stack.h
2 grammar.txt	8 parserDef.h	14 symbolTable.c
3 lexer.c	9 parseTree.c	15 symbolTable.h
4 lexer.h	10 parseTree.h	16 coding details.pdf
5 lexerDef.h	11 makefile	17 variables.c
6 parser.c	12 stack.c	18 variables.h

**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: Maximal Match.
- [B].Keyword Handling Technique: Whenever we get a lexeme from the DFA, we check in the symbol table that the lexeme is keyword or not.
- [C].Hash function description, if used for keyword handling:\_Implemented rolling hash function along with linear probing
- [D].Have you used twin buffer? (yes/ no) \_\_yes\_\_
- [E].Error handling and reporting (yes/No): \_\_yes\_\_
- [F].Describe the errors handled by you: 1) Unknown symbol, 2) Unknown Pattern, 3)If the identifier size is greater than maximum size or less than minimum size given in the grammar.
- [G].Data Structure Description for tokenInfo (in maximum two lines): It is a struct with 3 elements:  
1)tokenName, 2)a union which can have 3 values(lexemeValue, NUM, RNUM), 3)line number

**6. Parser Details:**

- [A].High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):
  - i. grammar : This is an array of the struct named Rule where the struct rule contains 3 elements:an array rule of type gSym where gSym is a structure which is basically stands for grammar symbol and states whether the grammar symbol is a terminal or non terminal and its enum value ,length of each rule in grammar, and Boolean synch. Each rule element consists of gSyms where the first element consists of a non terminal and the remaining consist of the grammar symbols it derives.
  - ii. FIRST and FOLLOW sets : First and Follow sets are both arrays of elements belonging to a structure called Set. The structure Set contains an array of elements of type gSym and the size of the array. Set basically refers to The array of terminals that belong to the first/follow set of a particular non terminal.

- iii. parse table Parse Table is a 2 dimensional array which has one dimension based on the size of non terminals and the other dimension based on the size of terminals and the structure contained is rules and thus it basically gives the rule which the the parser has to follow when it sees a particular terminal on the top of stack and a look-ahead token and expand using that rule.
- iv. parse tree: (Describe the node structure also) : The basic structure of a node of parse tree includes 5 fields namely leftmost which points to the leftmost child, rightmost which points to the rightmost child, parentNode which points to the parent node of the current node, nodeValue which saves a part of the rule in the node and token which gets initialized to the matching token of the stack during the traversal.
- v. Any other (specify and describe) : 1)gSym: stands for grammar symbol. It contains 3 items: a Boolean isTerminal, enum entry of Terminal, and enum entry of nonterminal. 2)

#### [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)  
The order which we have adopted is:-  
Lexeme Line-No tokenName parentNodeSymbol isLeafNode nodeSymbol

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

- i. Data structure for original grammar rules : Array of structs
- ii. FIRST and FOLLOW sets computation automated (yes /no): yes
- iii. Name the functions (if automated) for computation of First and Follow sets: The primary function for computing the first and follow sets is ComputeFirstAndFollow() this function calls the recursive function function first\_set() which thus computes the first set. After the first set is computed follow set is computed in the function ComputeFirstAndFollow() itself.
- iv. If computed First and Follow sets manually and represented in file/function (name that) \_\_\_\_\_

#### [D].Error Handling

- v. Attempted (yes/ no): yes
- vi. Describe the types of errors handled : We have implemented panic mode error recovery using synch sets. 1) If terminal on lookahead does not produce any rule with the non terminal on the top of stack, we move the lookahead forward and generate an "Invalid token" error. 2)If terminal on lookahead does not match terminal on top of stack, we generate an "Error" with corresponding line number. Lexical errors are printed separately.

#### 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/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): Executes correctly on all test cases, but some of the errors don't match for test case 6.
- [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

#### 10. Specify the language features your lexer or parser is not able to handle (in maximum one line) :Any lexeme can have a predefined max size which is hardcoded.

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

12. Declaration: We, Nek Manchanda, Utkarsh Sharma, Yash Sejpai, Nishant Singh, Prakhar Mundra 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

Name: Nek Manchanda ID: 2021A7PS0576P

Name: Utkarsh Sharma ID: 2021A7PS2429P

Name: Yash Sejpai ID: 2021A7PS2683P

Name: Nishant Singh ID: 2021A7PS2689P

Name: Prakhar Mundra ID: 2021A7PS2694P

Date: 05/03/2024

---

*Not to exceed 3 pages.*