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

Compiler Construction (CS F363)
II Semester 2024-25
Compiler Project
Coding Details
(March 15, 2025)

Group Number

46

1. Team Members Names and IDs
   ID 2022A7PS0293U Name Suryavir Kapur
   ID 2022A7PS0168U Name Ronit Dhansoia
   ID 2022A7PS0177U Name Anagh Goyal
   ID 2022A7PS0114P Name Harshwardhan Sugam

2. Mention the names of the Submitted files :

   1. `constants.h` 2. `dataStructures.h` 3. `grammar.h`
   4. `lexer.h` 5. `lexerDef.h` 6. `parse.h`
   7. `parser.h` 8. `parserDef.h` 9. `utils.h`
   10. `firstFollow.h 11. `dataStructures.c` 12. `driver.c`
   13. `firstFollow.c` 14. `grammar.c` 15. `lexer.c`
   16. `parse.c` 17. `utils.c` 18. `GRAMMAR.gmr`
   17. `makefile`

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : 18

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

5. **Lexer Details:**
   [A]. Technique used for pattern matching:
   - dfa using switch case statements and backtracking

   [B]. Keyword Handling Technique:
   - hashtable called Keyword Table

   [C]. Hash function description, if used for keyword handling:
   a. (b+33) + c
   b. custom base: 5187
   c. prime number

   [D]. Have you used twin buffers? (yes/ no): yes
   [E]. Error handling and reporting (yes/No): yes
   [F]. Describe the errors handled by you
   i. pattern and symbol matching
   ii. real number having back to back E (test case 2)
   iii. length errors according to regex in language specifications

   [G]. Data Structure Description for tokenInfo (in maximum two lines): name terminalMap enum, string, line number, is number and it's value

6. **Parser Details:**

[A]. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):
   i. grammar : pointer to rules -> likedlist of symbols -> containing symbols

ii. FIRST and FOLLOW sets:
   - 2d int arrays firstSet and FollowSet
   - rows: non-terminals, columns: terminals. '1' = in set, '0' = not in set.

iii. parse table:
   1. again 2d int arrays
   2. same as above, except rule number if present

iv. parse tree:
   1. parsetree based on a root narytreenode.
   2. narytreenode: leaf (terminal, token info) or non-leaf (non-terminal ID)

v. Any other (specify and describe) stack and narry tree setups and auxiliary functions

[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:
   - inorder traversal
   - dfs, left to right search

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

i. Data structure for original grammar rules: text file

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

iii. Name the functions (if automated) for computation of First and Follow sets: computeFirstAndFollowSets, which depends on computeFirstSets as that is calculated recursively

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

[D]. Error Handling

v. Attempted (yes/ no): yes

vi. Describe the types of errors handled:
   - syntax, errors, basic recovery via continuation popping of the stack
   - lexical errors, are printed directly by the parser, not passed in

7. Compilation Details:

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

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

[C]. Mention the .c files that do not compile: no

[D]. Any specific function that does not compile: no

[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):
   - lexer properly functioning, although it skips end character sometimes, we append two new lines to fix this
   - parser does work somewhat, but again only on the test cases provided by IC, not other wise

[B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: NA

10. Specify the language features your lexer or parser is not able to handle (in maximum one line)

- it has problems with unions
- it can't parse most of our own test cases
- lexer works perfectly though

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

12. **Declaration:** We, Suryavir, Ronit, Anagh and Harshwardhan 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

    Date: 14.03.25
    ID 2022A7PS0293U Name Suryavir Kapur
    ID 2022A7PS0168U Name Ronit Dhansoia
    ID 2022A7PS0177U Name Anagh Goyal
    ID 2022A7PS0114P Name Harshwardhan Sugam

    Not to exceed 3 pages.