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
2

---

1. Team Members Names and IDs

ID 2020B4A70833P	Name Aryan Barypatre
ID 2020B4A71654P	Name Bhuvan Gupta
ID 2020B4A70969P	Name Dhruv Singh
ID 2020B2A71932P	Name Pulkit
ID 2020B3A71958P	Name Jash Naik
ID 2020B2A71406P	Name Siripuram Surya

2. Mention the names of the Submitted files:

1 grammar.txt	7 parser.h	13 testcase3.txt
2 coding details.pdf	8 parser.c	14 testcase4.txt
3 lexerDef.h	9 driver.c	15 testcase5.txt
4 lexer.c	10 makefile	16 testcase6.txt
5 lexer.h	11 testcase1.txt	17 README.md
6 parserDef.h	12 testcase2.txt	

3. Total number of submitted files (including copy the pdf file of this coding details pro forma) : 17 (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: Implemented DFA using switch Case which runs over an infinite loop till some pattern matches or an error is encountered.
   - [B]. Keyword Handling Technique: Implemented a lookup Table where keywords were inserted using Hashing.
   - [C]. Hash function description, if used for keyword handling: The hash function takes string as an input along with its length and uses Horner's rule to compute a polynomial in x =16, in addition to that bitwise & and ^ operations are applied to reduce collisions and then final hash value is calculated by modulo Hash table size on computed hash value. Through experimentation on different hash table sizes ranging from 32 to 128, 70 was found to be most suitable considering collision frequency and memory consumption.
   - [D]. Have you used twin buffer? (yes/ no): yes
   - [E]. Error handling and reporting (yes/No): yes
   - [F]. Describe the errors handled by you: Length exceeded of function name and variable name, Unknown Symbol and Unknown pattern.
   - [G]. Data Structure Description for token Info (in maximum two lines): Created a structure Token storing line number, token id as an ENUM and lexeme in string format.

6. **Parser Details:**

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

i. grammar: Used hash table to store grammar rules where corresponding to each nonterminal hash value was generated which was used as a key and value is the 2D array of strings where each row represents the production rule generated by the corresponding nonterminal.

ii. FIRST and FOLLOW sets for each nonterminal first and follow set are stored using 1D array of strings with a feature that only unique elements present in the array.

iii. parse table Used hash table to store parse table where key is generated by applying hashing on non-terminals and value consist of array where each element corresponds to terminal storing the production rule from which nonterminal derives the terminal or null incase of error and TK_SYN in case of terminal in follow set.

iv. parse tree: (Describe the node structure also) Each node of the parse tree stores a string denoting nonterminal or terminal or lexeme besides that it contains array of pointers pointing to its child nodes in addition to a field storing number of child nodes and line number denoting current line number of token in input for error printing and recovery.

v. Any other (specify and describe) Used Doubly LinkedList to pass the tokens from lexer to parser which also acted as input tape for parser and Stack which is implemented using Doubly LinkedList. Used a List structure to store errors line no. wise so that errors could be reported in order.

[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) In order Traversal with Leftmost child --> parent node--> remaining siblings (excluding the leftmost)

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

i. Data structure for original grammar rules Hash table along with a 2D array of string as value of each key of hash table
ii. FIRST and FOLLOW sets computation automated (yes /no) yes
iii. Name the functions (if automated) for computation of First and Follow sets calculateFirst and calculateFollow

[D]. If computed First and Follow sets manually and represented in file/function (name that) Not Applicable

Error Handling

iv. Attempted (yes/ no): yes
v. Describe the types of errors handled Input Symbol not matching with Top of the Stack, Invalid Symbol, Invalid Token, Stack is empty but input tape of tokens is not and input tape is empty but stack is not empty.

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): Implemented lexer, automated first and follow set, parse table and parse tree with all testcases running correctly and reporting desired output.
   [B]. Gives segmentation fault with any of the test cases (1-6) uploaded on the course page. If yes, specify the testcase file name: None

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 (Yes/No): No

12. Declaration: We, Aryan Barapatre, Bhuvan Gupta, Dhruv Singh, Pulkit, Jash Naik, Siripuram Surya (your names) 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 2020B4A70833P	Name Aryan Barypatre
ID 2020B4A71654P	Name Bhuvan Gupta
ID 2020B4A70969P	Name Dhruv Singh
ID 2020B2A71932P	Name Pulkit
ID 2020B3A71958P	Name Jash Naik
ID 2020B2A71406P	Name Siripuram Surya

Date: 5/3/24

Not to exceed 3 pages.