

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)

Group No.

42

1. IDs and Names of team members

ID: 2020A7PS1214P Name: Darshan Abhaykumar

ID: 2020A7PS0970P Name: Debjit Kar

ID: 2020A7PS0986P Name: Nidhish Parekh

2. Mention the names of the Submitted files:

grammar.txt
driver.c
makefile
lexer.c
tree.c
lexer.h
grammar.c
stack.h
stackDef.h
testcase1.txt
testcase2.txt
testcase3.txt
testcase4.txt
testcase5.txt
testcase6.txt
testcase7.txt
lexerDef.h
parser.c
linkedList.h
HTable.c
grammar.h
parser.h
linkedList.c
grammarDef.h
parserDef.h
stack_ll.c
hashtable.h
stack.h
tree.h
treeDef.h
NlinkedList.c
NlinkedList.h
NlinkedListDef.h
README.md

3. Total number of submitted files: **33** (All files should be in **ONE folder** named exactly as Group_#)
4. Have you mentioned your names and IDs at the top of each file (and commented well)? **Yes**
5. Have you compressed the folder as specified in the submission guidelines? **Yes**

6. Lexer Details:

- [A]. Technique used for pattern matching: Based on input, state transitions are made in the DFA
- [B]. DFA implementation (State transition using switch case, graph, transition table, any other (specify):
Switch for state transition along with retraction function and error reporting
- [C]. Keyword Handling Technique: Look-up table
- [D]. Hash function description, if used for keyword handling: Converts string to a hashed key (Linear Probing)
- [E]. Have you used twin buffers? **Yes**
- [F]. Lexical error handling and reporting: **Yes**
- [G]. Describe the lexical errors handled by you: **1)** Unknown character at start state
2) Invalid pattern detected
3) Integer size too big
4) Reached EOF
5) Length of string too long
- [H]. Data Structure Description for tokenInfo (in maximum two lines):
Enum token type, line number, Union struct analyses the data types inside and efficiently computes the required size
- [I]. Interface with parser : parser calls getNextToken() in the parseInputSourceCode() function

7. Parser Details:

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

- i. grammar: Array of array of linked lists
- ii. parse table
- iii. parse tree: (Describe the node structure also)
- iv. Parsing Stack node structure:
- v. Any other (specify and describe)

[B]. Parse tree

- i. Constructed: **No**
- ii. Printing as per the given format (yes/no): **NA**
- iii. Describe the order you have adopted for printing the parse tree nodes (in maximum two lines) : **NA**

[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 : **No**
- iii. Data structure for representing sets: Array of array of linked lists

- iv. Time complexity of computing FIRST sets: $O(m*n)$
m: max no. of terms in a rule on RHS **n**: max no. of rules of a non-terminal
- v. Name the functions (if automated) for computation of First and Follow sets: NA
- vi. If computed First and Follow sets manually and represented in file/function
computeFirsts() and computeFollows() in parser.c

[D]. Error Handling

- i. Attempted: Yes
 - ii. Printing errors: One at a time
 - iii. Describe the types of errors handled
Terminal mismatch, Input consumed but stack is not empty,
Input is remaining but stack is empty, table entry is NULL
 - iv. Synchronizing tokens for error recovery (describe):
 - v. Total number of errors detected in the given testcase t6(with_syntax_errors).txt
-

8. Compilation Details:

- [A].makefile works: Yes
- [B].Code Compiles: Yes
- [C].Mention the .c files that do not compile: _____
- [D].Any specific function that does not compile: _____
- [E].Ensured the compatibility of your code with the specified gcc version: Yes

9. Driver Details: Does it take care of the options specified earlier(yes/no): _____

10. Execution

[A].status (describe in maximum 2 lines): _____

[B]. Execution time taken for

- testcase1.txt (in ticks) **17481** and (in seconds) **0.017481**
- testcase2.txt (in ticks) **13513** and (in seconds) **0.013513**
- testcase3.txt (in ticks) **16198** and (in seconds) **0.016198**
- testcase4.txt (in ticks) **22296** and (in seconds) **0.022296**
- testcase5.txt (in ticks) **117413** and (in seconds) **0.117413**
- testcase6.txt (in ticks) **114018** and (in seconds) **0.114018**
- testcase7.txt (in ticks) **15962** and (in seconds) **0.015962**

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

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

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

13. Declaration: We, **Darshan Abhaykumar, Debjit Kar and Nidhish Parekh**, 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.

ID: 2020A7PS1214P Name: Darshan Abhaykumar

ID: 2020A7PS0970P Name: Debjit Kar

ID: 2020A7PS0986P Name: Nidhish Parekh

Date: 3rd March 2023

Should not exceed 4 pages.