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

Batch No. :

**DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS**

**Compiler Construction (CS F363)**

**II Semester 2017-18**

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

**Coding Details**

**(February 26, 2018)**

1. **Personal details**

ID 2015A7PS0102P

Name K S Sanjay Srivastav

1. **Files and folder details**
2. Mention the names of the Submitted files :

1 codingDetails.docx 7 parser.c 13 testcase3.txt

2 driver.c 8 parser.h 14 testcase4.txt

3 lexer.c 9 parserDef.h 15 testcase5.txt

4 lexer.h 10 grammar.txt

5 lexerDef.h 11 testcase1.txt

6 makefile 12 testcase2.txt

1. Total number of submitted files:15
2. Have you compressed folder as specified in the submission guidelines? yes
3. **Lexer Details:**
   1. Technique used for pattern matching: Each state in the DFA is assigned a number and based on that number, the program flow changes using switch case statements.
   2. Keyword Handling Technique: A global hash table is used to match the tokens (if token is ID or FUNID) with the keywords.
   3. Hash function description, if used for keyword handling: The hash function

takes a string and size of hash table and based on the index of the character in the string, a polynomial is constructed (with x=33). Finally it brings down to the hash table size (i.e. 0 to hashTable size -1).

* 1. Have you used twin buffer? (yes/ no) yes
  2. Error handling and reporting (yes/No):yes
  3. Describe the errors handled by you: Errors will be reported in case the size criteria is not properly met (also if twin buffer cannot handle it , buffer size is 30 charcters each). Also errors like unrecognised characters are properly incorporated.
  4. Data Structure Description for tokenInfo (in maximum two lines): A token has a tokenid, its value and a line number. A token node has pointer to a token and a pointer to its next. A tokenlist structure is maintained for linked list’s head and tail.

1. **Parser Details:** 
   1. High Level Data Structure Description (in maximum three lines each, avoid giving C definitions used):
      1. grammar :Grammar is an array of pointers to nodes, where each node has a number, isTerminal flag (the number and flag uniquely determines terminals and non-terminals) It also has a head pointer to an array of linked lists (nodeList) of such nodes each linkedlist corresponding to a rule in the grammar. Node also has the size of the array.
      2. parse table: A 2-D array of pointers to rules in the grammar. Each index represents the non-terminal ,terminal pair.
      3. parse tree: (Describe the node structure also). It is an n-ary tree. A parsetree node has a Grammar node pointer, a token node pointer (lexer’s token), a children pointer, a next pointer (sibling) and a par pointer (parent). Every level in the tree is a linkedlist of such nodes, every nonterminal has a children node and every terminal has a token node. Grammar node pointer has this information of whether it is a terminal or non-terminal. For error recovery,a special node is used called nullNode ( for non-terminals children)
      4. Any other (specify and describe) A Stack data structure having a parse tree node pointer as its element. So in order to update the line number and the token information, while popping a terminal from stack, I directly update the token for the parse tree node, without traversing the tree as pointer to it is stored in stack.
   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) As mentioned in the specifications, it is output to a file. It has 7 columns in the order : lexemeCurrentNode lineno token valueIfNumber

parentNodeSymbol isLeafNode(yes/no) NodeSymbol. It first prints the left child node, the parent node and all its other child nodes in order.

* 1. Computation of First and Follow Sets
     1. Data structure for First and Follow sets : Unsigned int arrays.
     2. FIRST and FOLLOW sets computation automated (yes /no) yes.
     3. Name the functions (if automated) for computation of First and Follow sets: getFirstSet, FIRST, getFollowSet.
     4. If computed First and Follow sets manually and represented in file/function : -
  2. Error Handling and recovery
     1. Attempted (yes/ no): yes
     2. Synchronizing set formation details: Synchronizing set of each non-terminal has the follow set of it stored.
     3. Describe the types of errors handled Syntax. When a token on top of stack is a terminal and is not matched with input, an error is raised and the token is skipped along with the popping of stack top. When a nonterminal on top of stack has NULL entry in its M[X][a] then the input is skipped till it is in the synch set of it. Then the non-terminal is popped and continued. Also if a rule derives eps, it is popped if the next token is not matched.

1. **Compilation Details**
   1. Makefile works (yes/no): yes
   2. Code Compiles (yes/ no): yes
   3. Mention the .c files that do not compile: -
   4. Any specific function that does not compile: No
   5. Ensured the compatibility of your code with the specified gcc version: yes
2. **Driver Details:** Does it take care of the options specified earlier(yes/no):yes
3. **Execution details**
   1. status (describe in maximum 2 lines): Code is working fine with all error checkings done. Lexical Errors are first displayed follow by syntax errors.
   2. Gives segmentation fault with any of the revised test cases (1-5) uploaded on the course page. If yes, specify the testcase file name: -
4. Specify the language features your lexer or parser is not able to handle (in maximum one line) -
5. **Lifeline detail:** Are you availing the lifeline (Yes/No): No
6. **Declaration**: I, K S Sanjay Srivastav (your name) declare that I have put my genuine efforts in creating the compiler project code and have submitted the code developed only by me. I have not copied any piece of code from any source. If my code is found plagiarized in any form or degree, I understand that a disciplinary action as per the institute rules will be taken against me and I will accept the penalty as decided by the department of Computer Science and Information Systems, BITS, Pilani.

ID 2015A7PS0102P Name: K S Sanjay Srivastav Date: 26-02-2018