Batch No. : 39

Group No. - 39

BIRLA INSTITUTE OF TECHNOLOGY AND SCIENCE, PILANI

DEPARTMENT OF COMPUTER SCIENCE AND INFORMATION SYSTEMS

Compiler Construction (CS F363)
II Semester 2018-19
Compiler Project (Stage-2 Submission)
Coding Details
(April 14, 2019)

Instruction: Write the details precisely and neatly. Places where you do not have anything to mention, please write NA for Not Applicable.

1. IDs and Names of team members
   ID:_2016A7PS0026P___Name:___AKANKSHYA MISHRA_

   ID:_2016A7PS0034P___Name:___NARAPAREDDY BHAVANA___

   ID:_2016A7PS0052P___Name:___KARABEE BATTA___

   ID:_2016A7PS0062P___Name:___AASTHA KATARIA___

1. Mention the names of the Submitted files (Include Stage-1 and Stage-2 both)

1 lexer.c	7 AST.c	13 typechecker.h	19 DFA.c
2 lexer.h	8 ASTDef.h	14 typechecker.c	20 codegenDef.h
3 lexerDef.h	9 AST.h	15 semanticAnalyzer.h	21 driver.c
4 parser.c	10 SymbolTable.c	16 semanticAnalyzer.c	22 grammar.c
5 parser.h	11 SymbolTable.h	17 codegen.h	23 grammar.h
6 parserDef.h	12 SymbolTableDef.h	18 codegen.c	24 grammarDef.h
25 key.h	28 stack.h	31 makefile	34 testcase3.txt
26 key.c	29 stack.c	32 testcase1.txt	35 main1.txt

27 keyDef.h	30 stackDef.c	33 testcase2.txt	36 main2.txt
37 main3.txt	39 DFA.pdf		41 rules.txt
38 main4.txt	40 Semantic rules for AST.pdf	42 testcase3.txt	

43.testcase4.txt 44.testcase5.txt 45 testcase6.txt 46.testcase7.txt

47. testcase8.txt 48.testcase9.txt 49.testcase10.txt

1. Total number of submitted files: **41** (All files should be in ONE folder named exactly as Group_#, # is your group number)
2. Have you compressed the folder as specified in the submission guidelines? (yes/no)___**YES**__

1. Status of Code development: Mention 'Yes' if you have developed the code for the given module, else mention 'No'.
    1. Lexer (Yes/No: **yes**
    2. Parser (Yes/No): **yes**
    3. Abstract Syntax tree (Yes/No): **yes**
    4. Symbol Table (Yes/ No): **yes**
    5. Type checking Module (Yes/No): **yes**
    6. Semantic Analysis Module (Yes/ no): **yes** (reached LEVEL as per the details uploaded)
    7. Code Generator (Yes/No): No
2. Execution Status:

1. Code generator produces code.asm (Yes/ No): **No**

2. code.asm produces correct output using NASM for testcases (Main#.txt, #:1-4): **No**

3. Semantic Analyzer produces semantic errors appropriately (Yes/No): **Yes**

4. Type Checker reports type mismatch errors appropriately (Yes/ No): **Yes**

5. Symbol Table is constructed (yes/no) **yes** and printed appropriately (Yes /No): **yes**

6. AST is constructed (yes/ no) **yes** and printed (yes/no) **yes**

7. Name the test cases out of 7 as uploaded on the course website for which you get the segmentation fault (testcase#.txt ; # 1-3 and Main@.txt ; @:1-4):**__None__________**

3. Data Structures (Describe in maximum 2 lines and avoid giving C definition of it)

   1. AST node structure**__________union of leaf and non-leaf nodes, parent node, and type of ASTNode**

   2. Symbol Table structure :

   3. Data structure for global variables: **Global hash table which stores both record data type and global variables**

   4. Record type expression structure: **stored in global hash table which has pointers to link list of field data type**

   5. Input parameters type structure:**_data types of input parameters stored in link list attached to hash element of outer symbol table**

   6. Output parameters type structure:**__ data types of output parameters stored in link list attached to hash element of outer symbol table**

7. Structure for maintaining the three address code(if created : **quadruple (struct)**

8.Any other interesting data structure used: **Quadruples and their traversal using quad head and quad tail**

1. Semantic Checks: Mention your scheme NEATLY for testing the following major checks

   1. Variable not Declared: **if not found in global and function symbol table(hashing)**
   2. Multiple declarations: **if already present in either of the hash tables(hashing)**
   3. Number and type of input and output parameters: **matching of link lists**
   4. assignment of value to the output parameter in a function : **tag used in link list of output parameters**
   5. function call semantics: **comparing link lists**
   6. type checking: **hashing into the symbol table**
   7. return semantics: **comparison of link lists (hashing)**
   8. Recursion: **hashing**
   9. function overloading: **hashing**
   10. 'while' loop semantics: **tags, checking if the identifier is assigned by a function call or read statement or assignment statement**
   11. record data type semantics and type checking of record type variables: **hashing in the global symbol table and checking the link list**

   12. register allocation: **Not implemented**

   13. The scope of variables and their visibility: **use of global and function hash tables**

1. Compilation Details:

   1. Makefile works (yes/No): **yes**

   2. Code Compiles (Yes/ No): **yes**

3. Mention the .c files that do not compile:__Codegen.c (code generator leads to segmentation fault) ; intermediate code generator works fine_

4. Any specific function that does not compile: none
5. Ensured the compatibility of your code with the specified gcc version(yes/no)__yes

1. Driver Details: Does it take care of the options specified earlier?(yes/no):**yes**

2.. Specify the language features your compiler is not able to handle (in maximum one line): none

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

1. Write exact command you expect to be used for executing the code.asm using NASM simulator:-

1. Strength of your code(Tick the boxes where applicable): (a) correctness - **yes** (b) completeness - **yes** (c) robust - **yes** (d) Well documented - **yes** (e) readable - **yes** (f) strong data structure - **yes** (g) Good programming style (indentation, avoidance of goto stmts etc) - **yes** (h) modular - **yes** (i)space - **yes** and time efficient **-yes**

2. Any other point you wish to mention: _ **none**

1. Declaration: We, **Akankshya Mishra, Narapareddy Bhavana, Karabee Batta, and Aastha Kataria** declare that we have put our genuine efforts in creating the compiler project code and have submitted the code developed 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 us and we will accept the penalty as decided by the Department of Computer Science and Information Systems, BITS, Pilani.

Date:April 14,2019
(Not to exceed beyond 3 pages)