

ASSIGNMENT 1

CONSTRUCTION OF DAG & THREE ADDRESS CODE

May 30, 2017

1 Introduction

In this assignment you have to construct Directed Acyclic Graph (DAG) and Three Address Code (TAC) for simple expressions. At first, you have to download the stub code given in moodle. The provided lex and yacc file can recognize simple expression consists of $+$, $-$, $*$, $/$, $()$, variables and integer literals. There is another file named **dag.cpp**. This file contains a class definition and some function prototypes. Your task is to write appropriate code in this file.

2 Tasks

You have to complete the following tasks in this assignment.

2.1 Constructing DAG

The provided yacc file contains grammar and semantic actions that resembles the SDD given in the textbooks at page 360. We will construct DAG using that SDD and value number method. The DAG will be stored in an array named `dag_array` which is defined in `dag.cpp` file. Indexing will start from 1 in this array. A node in a dag will contain a label and two integers representing indices of left child and right child in the array. For a leaf node, the two child indices will be 0. The **constructNode** function will contain the implementation of value number method. You have to do hashing to implement value number method efficiently. You may find `unordered_map` of standard template library helpful in this regard.

You will also need to write code for printing the `dag_array` in the **print_dag_array** function. This function is called from the first production of our grammar. That means we will print the

dag_array after successful parsing of input and construction of DAG for the input.

2.2 Constructing TAC

After constructing DAG, you will have to produce Three-Address Code from the DAG. You will only have to print the Three-Address Code in the output file. There is no need to save it in any data structure like quadruple or triple. You may find the figure 6.4 of the textbook helpful in this regard.

3 Input

The input will be a text file containing an expression in a single line.

4 Output

You have to produce an output file named **output.txt**. This file will contain the DAG array and the Three-Address Codes for the input expression. You have to follow the sample output file strictly.

You will also have to write the method of your solution in a text file named **solution.txt**. This file is not an output of the written program, rather you will write one or two small paragraphs stating your approach to solve the assignment.

5 Rules

- **Plagiarism is strongly prohibited.**
- No submission after the deadline will be allowed.
- Deadline will not extend in any situation.

6 Submission

All Submission will be taken via moodle. Follow the steps given below to submit your assignment.

1. In your local machine create a new folder whose name is your 7 digit student id.
2. Put the scanner.l, parser.y, dag.cpp and solution.txt file in the folder you have just created. **Don't** rename these files. You can also put a script to compile and run your program. Do not put any lex.yy.c, y.tab.c, y.tab.h, y.output, a.exe files in this folder.

3. Compress the folder in a **zip** file which should be named as your 7 digit student id. The extension of the submitted file should be **zip**. Any other extension will result in reduction in marks.
4. Make sure that after extracting the zip file, only a folder containing your roll number is produced.
5. Submit the zip file within the Deadline.

7 Marking

- **Construction of DAG:** 45%
- **Construction of TAC:** 35%
- **Explanation of solution:** 10%
- **Proper Submission:** 10%

8 Deadline

Submission deadline is set at **3:00 pm** of **7 June, 2017**.