Assignment-1

Dasari Shree Ujjwal - EE18BTECH11010

Github repository

https://github.com/dsujjwal/C−DS

1 Problem

Consider the following C Code Segment:

```
a = b + c ;
e = a + 1 ;
d = b + c ;
f = d + 1 ;
g = e + f ;
```

In a compiler, this code segment is represented internally as a directed acyclic graph (DAG).The number of nodes in the DAG is

2 Solution

Answer: 6.

DAG is a useful data structure for implementing transformations on **Basic Blocks**. A Basic Block is a straight line code sequence which has no branches in and out branches except to the entry and at the end respectively.It is a set of statements that always executes in a sequence one after the other. The code segment in the question is an example of a Basic Block. A DAG is usually constructed using **Three-Address Codes**.

General form of Three-Address Code:
Three-address code is an intermediate code used by optimizing compilers to aid in the implementation of code-improving transformations. The general form of a Three-Address Code is as follows:

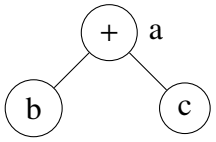
a = b op c

Here,
1) **a,b,c** are the operands which are constants,names or compiler generated temporaries.
2) **op** represents operator

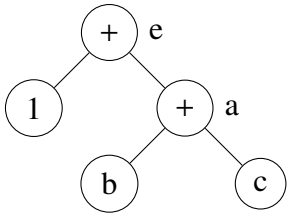
Rules for construction of DAGs:
Rule 1: In a DAG,Interior nodes always represent the operators. Exterior nodes (leaf nodes) always represent the names, identifiers or constants.
Rule 2: While constructing a DAG, a check is made to find if there exists any node with the same value. A new node is created only when there does not exist any node with the same value.This action helps in detecting the common sub-expressions and avoiding the re-computation of the same.
Rule 3: The assignment instructions of the form x:=y are not performed unless they are necessary.

Construction of DAG:
1) The first step is drawing two nodes representing b and c (Rule 1).

2) Now,we connect node b and node c to a new node $'+'$, completing the representation of Statement 1 of code (Rule 1 and 2).Since the node $'+'$ represents a , we mention a beside the node $'+'$.

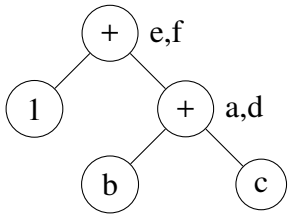


3) Now,we make a new node 1 and connect the new node 1 and $'+'$ (representing a) to a new node $'+'$ (representing e), completing the representation of Statement 2 of code (Rule 1 and 2).Since the node $'+'$ represents e , we mention e besides the node.

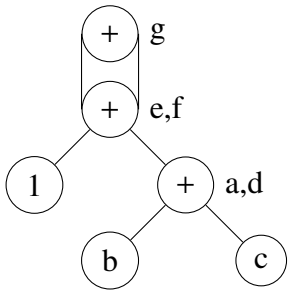


4) Statement 3 of the code is equivalent to Statement 1.So we don't make a new node for d and just mention it beside a.Similarly Statement 4 of the

code is equivalent to Statement 2.So we don't make a new node for f and just mention it beside e.



5) Now as e and f are same, we make a new node g and connect it to $'+'$ (representing e and f) using two edges. With this we have completed the representation of the complete code.



Now can clearly see that the total number of nodes in the DAG is 6.