

CS 536

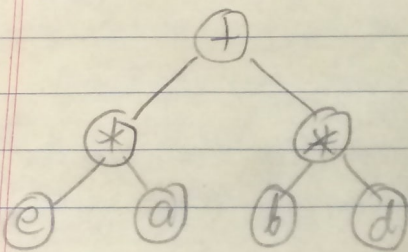
HW-10

SPARSH AGARWAL

9075905142

A1) (i)  $e * a + b * d$  (will require 3 registers)

AST

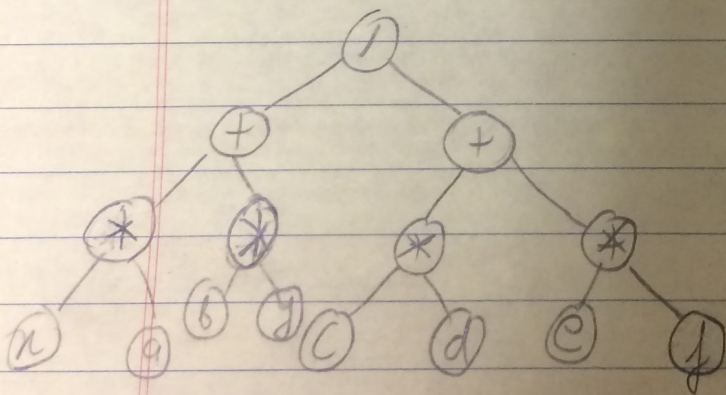


Pseudo code

load a into T0  
load e into T1  
 $T0 = T0 * T1$   
load b into T1  
load d into T2  
 $T1 = T1 * T2$   
 $T0 = T0 + T1$

(ii)  $(x * a + b * y) / (c * d + e * f)$  (will use 4 registers)

AST



load b into T1  
load y into T2  
 $T1 = T1 * T2$   
 $T0 = T0 + T1$

Pseudo code

load x into T0  
load a into T1  
 $T0 = T0 * T1$   
load c into T1  
load d into T2  
 $T1 = T1 * T2$   
load e into T2  
load f into T3  
 $T2 = T2 * T3$   
 $T1 = T1 + T2$   
 $T0 = T0 / T1$



A2)

```
int numRegister (ASTNode node) {  
    if (node.isleaf()) {  
        return 1; }  
    if (node.leftChild.isleaf() || node.rightChild.isleaf()) {  
        return numRegister(node.leftChild) + numRegister(node.rightChild);  
    } else {  
        return numRegister(node.leftChild) + numRegister(node.rightChild) - 1;  
    }  
}
```

(how does associativity & commutativity affect?)

A2)

```
int numRegister (ASTNode node) {  
    if (node.isleaf()) {  
        return 1;  
    }  
    left = numRegister (node.leftChild);  
    right = numRegister (node.rightChild);  
    if (left == right) {  
        return left + 1;  
    }  
    if (left > right) {  
        return left;  
    }  
    if (left < right) {  
        return right;  
    }  
}
```