HW2

21300506 LEE KI HYUN

1. Problem Definition

Input ​ : statement of string form (standard input) (ex) (or a1 (not (or (not (or a2 a3)) a4))) Output ​ : n value of the statement “an” (ex) 1 \n 2 -4 \n 3 -4 Receive logic from user in statement form, and use recursion to make input to binary tree form. If it is a valid expression, create NNF through De Morgan’s Law and change NNF to DNF, which is disjunction of conjunction form through distributed law. After that, convert the statement ‘or’ to ‘\n’, ‘and’ to ‘\t’, ‘not’ to ‘-’, and n of ‘an’ to integer, and then print tree.

1. Approach

The structure ​Tree ​is defined to have 2 contents (sign and prop), and 2 pointers (left and right). The integer sign is used to identifying a node, 0(proposition),1(and),2(or),3(not). The integer prop the number of a proposition, so it is used only when the sign is 0. There are several functions for the structure​ Tree.

NNF -

The function​ NNF ​ makes the tree to be without the node of ​not. It calls the function ​demorgan ​if the first node of the tree is node ​not. It calls itself(NNF) otherwise. It returns NULL if the first node has NULL pointer; return the pointer of input​ t ​otherwise.

DNF -

The function DNF makes the tree to DNF form. It calls the function ​distribute ​if the first node of the tree is a node ​and(conjunction). Then, it calls DNF again for left and right nodes. It calls itself DNF if the first node of the tree is not a node ​and(conjunction). At the last, It returns the pointer of input​ t .

3) Result

In case input is “(or a1 (not (or (not (or a2 a3)) a4)))”, we can get a output “1 \n 2 -4 \n 3 -4”.

1. Discussion

Because I solved this problem using binary tree, there is a problem in some cases where there are more than three variables in an operator. And I have trouble in getting solution for this problem.