Discrete Mathematics HW2 : Converting a formula

21800412 HeeJu Shin

Introduction

- **Summary**
- **Problem analysis**

- **Problem** is Implement C program changing input (propositional formula) to DNF and finding out answer of formula. **Input** is a positional formula consisting of parentheses, 'or', 'and', 'not' and a n (n =positive integer). **Output** consists of DNF representation, 0, and solution of formula. 'Negation' is expressed in minus. **Analysis** of DNF - Disjunctive Normal Form, a conjuctive or disjunction of multiple conjuctive clauses.

Approach

- **Solution design**

  Get proposition and divide and insert it into the tree. Convert to NNF, convert to DNF. Print the answer using proposition converted to DNF.

- **Top-down description solution**

  1. Make initial formula

     Create an empty node and insert the formula received as an input. Remove the parentheses of formula, cut it with 'not', 'and', 'or' 'a n ', and put it in the binary tree. If key of current node is ‘and’ or ‘or’, node have left child and right child. If key of current node is ‘not’, node have left child. If key of current node is a n , node is leaf node. I changed ‘not’, ‘and’ and ‘or’ to ‘n’, ‘a’, ‘o’ respectively for convenience.
  2. Implementation of NNF

     Using De Morgan’s laws and pre-order traversal.

     If key of current node is ‘not’ and key of next node is ‘and’ or not’, remove current node and change the key of next node ‘and’ to ‘or’ and ‘or’ to and. In addition, inssert ‘not’ before the left_child and right_child. If the key of current node is ‘not’ and next node is ‘not’, two nodes are offset. Just delete two nodes.
  3. Implementation of DNF

     Using pre-order traversal. Only for case for key of current node is ‘and’ and key of the next node is ‘or’. Save the address of node and link parent and next node. Then make new left nd right node that will be attached to ‘or’ node using current node. Lastly, link new nodes to ‘or’ node and change the address of node to the address of ‘or’ node. Apply theses rules recursively.
  4. Print the result and answer

     Print leaf nodes of performed DNF as result using pre-order traversal. For case of answer of formula, make print answer using DNF value. Get the number of answer using number of results. You only need to consider one of answers. Print out the value meet first condition and then freely print out the rest.

Result

The leaf nodes of the DNF formula tree is printed. The values bound by 'and' are separated by space, values bound by 'or' are separated by Enter, and the DNF result and answer are separated by zero. The last line represents one case of the answer.

Manual

It is described in the README file.

Discussions

- This homework become the chance of reviewing for concept of tree. I realized that I could easily express and modify the positional logic using data structure. My thoughts have broadened. I thought I'm able to make a simple and accurate implementation using c++.
- While looking into DNF and CNF, I happened to read a post. It was post talking about MIT SAT/SMT camp. There's problem 'Slither Link' in the post. In the problem, numbers represent the number of line around itself. I felt it is mix of PA1's Z3 problem and PA2's recursion problem. I tried to solve the second problem 'fill-a-pix' of PA1 using type of bool instead of int. However it can not solve all cases. I didn't know why. There was not much time at that moment, so I hurriedly worked out the logic and quickly moved to coding. But when I read this post, I realized the importance of making accurate logic. If the person who solved this problem had used DNF, not CNF, he would have been lost. I want to logically solve the problem of discrete mathematics which has not fully solved and this slither link problem. http://rosaec.snu.ac.kr/meet/file/20110626p2.pdf