JIE MEI z5173405

Question 1

Assume the string have n symbols, and n-1 operations between them. We can apply divide and conquer strategy. Counting both the number of placements of the brackets such that the subexpression evaluates true(T) and the number of ways it evaluates to false(F). For example, the given string is “false or true nor true”, T(1,2) would be the number of ways of making “false or true” evaluate to true with correct bracketing(in this case, T(1,2) = 1).

Otherwise, for each subproblems, we "split" the operator **m** around the expression so that everything on the left side of the operator is in its own parenthesis, and everything on the right side of the operator is in its own In parentheses to form two smaller expressions.

For example, splitting the sample expression into "nor" will result in "(false or true) nor (true)". Then, we evaluate each subproblem on both sides and combine the results together, depending on the type of operator we want to split, and whether we want the result to be evaluated as true or false. We solve both subproblems in parallel:

$$T(l,r) = \sum_{m=l}^{r-1} TSplit(l,m,r)$$

$$F(l,r) = \sum_{m=l}^{r-1} FSplit(l,m,r)$$

$TSplit(l,m,r)$

- $T(l,m) * T(m+1,r)$ if the operator m is ‘and’
- $T(l,m) * F(m+1,r) + T(l,m) * T(m+1,r) + F(l,m) * T(m+1,r)$ If the operator m is ‘or’
- $T(l,m) * F(m+1,r) + F(l,m) * T(m+1,r) + F(l,m) * F(m+1,r)$ If the operator m is ‘nand’
- $F(l,m) * F(m+1,r)$ If the operator m is ‘nor’

$FSplit(l,m,r)$

- $F(l,m) * F(m+1,r)$ if the operator m is ‘or’
- $T(l,m) * F(m+1,r) + F(l,m) * F(m+1,r) + F(l,m) * T(m+1,r)$ If the operator m is ‘and’
- $T(l,m) * F(m+1,r) + F(l,m) * T(m+1,r) + T(l,m) * T(m+1,r)$ If the operator m is ‘nor’
- $T(l,m) * T(m+1,r)$ If the operator m is ‘nand’

In different ranges l and r can be covered, each needs evaluation of TSplit or FSplit.