COMP3121 Assignment 3 - Q1

Demiao Chen z5289988

July 26, 2021

Acknowledgement: the answer is inspired from tutorial 4 question 5 solution.

Answer

Let there be n symbols, and $n-1$ operations between them. We introduce the following symbols:
$T(i,j)$ represents the number of ways to parenthesize the symbols between i and j (both inclusive) such that the subexpression between i and j evaluates to true.
$F(i,j)$ represents the number of ways to parenthesize the symbols between i and j (both inclusive) such that the subexpression between i and j evaluates to false.
So we need to solve $T(1,n)$, that is the the number of ways one can put parentheses in the expression with n symbols such that it will evaluate to true. To solve $T(1,n)$, we need to solve the two sub problems: $T(i,j)$ and $F(i,j)$.
The base case is that $T(a,a)$ is 1 if symbol a is true, and 0 if symbol a is false. The reverse applies to $F(a,a)$. Otherwise, we split sub problems around an operator k, so everything in the left and right of the operator is in its own brackets. According to the truth table of 'AND', 'OR', 'NAND', 'NOR', we solve the sub problems in the following way

$$T(i,j) = \sum_{k=i}^{j-1} \begin{cases} T(i,k) * T(k+1,j) \\ \qquad \text{if operator } k \text{ is 'AND'} \\ T(i,k) * F(k+1,j) + T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad \text{if operator } k \text{ is 'OR'} \\ T(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) + F(i,k) * F(k+1,j) \\ \qquad \text{if operator } k \text{ is 'NAND'} \\ F(i,k) * F(k+1,j) \\ \qquad \text{if operator } k \text{ is 'NOR'} \end{cases}$$

$$F(i,j) = \sum_{k=i}^{j-1} \begin{cases} T(i,k) * F(k+1,j) + F(i,k) * F(k+1,j) + F(i,k) * T(k+1,j) \\ \qquad \text{if operator } k \text{ is 'AND'} \\ F(i,k) * F(k+1,j) \\ \qquad \text{if operator } k \text{ is 'OR'} \\ T(i,k) * T(k+1,j) \\ \qquad \text{if operator } k \text{ is 'NAND'} \\ T(i,k) * T(k+1,j) + F(i,k) * T(k+1,j) + T(i,k) * F(k+1,j) \\ \qquad \text{if operator } k \text{ is 'NOR'} \end{cases}$$

After solving all sub problems, we can get the answer of the question from $T(1,n)$.

The time complexity is $O(n^3)$. There are $O(n^2)$ different ranges that i and j could cover, and each needs to evaluate $T(i,j)$ or $F(i,i)$ up to $n-1$ different spilt points. So the time complexity is $O(n^2 * n) = O(n^3)$