Mubarra Khan Z5161249 Tuesday, 28 July 2020

Q1

You are given a boolean expression consisting of a string of the symbols true, false, separated by operators AND, OR, NAND and NOR but without any parentheses. Count the number of ways one can put parentheses in the expression such that it will evaluate to true. (20 pts)

Solution: let there be n symbols, and n-1 operations between them. We solve the following two subproblems:

1. How many way are there to make the expression starting from the i^{th} symbol and ending at the r^{th} that evaluate to true (T).
2. How many ways to do the same but value to false (F) for example in *“true and false NAND true”*

Base case: $T(i,i)$ is 1 is symbol i is true, and 0 if symbol i is false. The reverse applies to $F(i,i)$

Recursion:

Otherwise for each subproblem, we ‘split’ the expression around an operator *m* so that everyone got the left of the operator is in its own bracket, and everything to the right of the operator is in its own bracket to form two smaller expressions.

- E.g. splitting the sample expression ‘*NAND’* would give “(true and false) nand (true)”.

We then evaluate the subproblems on each of the two sides, and combine the result together depending on the type of operator we are splitting by, and whether we want the result to evaluate to true or false. We solve both subproblems in parallel:

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

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

$$TSplit(l, m, r) = \begin{cases} T(l,m) \times T(m+1,r) & \text{if operator m is 'and',} \\ T(l,m) \times F(m+1,r) + T(l,m) \times T(m+1,r) + F(l,m) \times T(m+1,r) & \text{if operator m is 'or'} \\ T(l,m) \times F(m+1,r) + F(l,m) \times F(m+1,r) + F(l,m) \times T(m+1,r) & \text{if operator m is 'nand'} \\ F(l,m) \times F(m+1,r) & \text{if operator m is 'nor'} \end{cases}$$

$$Fsplit(l, m, r) = \begin{cases} T(l,m) \times F(m+1,r) + F(l,m) \times F(m+1,r) + F(l,m) \times T(m+1,r) & \text{if operator m is 'and'} \\ F(l,m) \times F(m+1,r) & \text{if operator m is 'or'} \\ T(l,m) \times T(m+1,r) & \text{if operator m is 'nand'} \\ T(l,m) \times F(m+1,r) + T(l,m) \times T(m+1,r) + F(l,m) \times T(m+1,r) & \text{if operator m is 'nor'} \end{cases}$$

Note that the equations inside the Tsplit and Fsplit functions are chosen to correspond with the truth tables of the corresponding operator.

The complexity is $O(n^3)$. There are n^2 different ranges that l and r could cover, and each needs the evaluations of TSplit or Split at up to n different splitting points.