Soham Chatterjee
Email: sohamc@cmi.ac.in
Course: Parallel Algorithms and Complexity

Assignment - 2
Roll: BMC202175
Date: August 18, 2023

Problem 1

Prove that in NC^1 circuits and formulas are equivalent.

Solution: Let $C \in NC^1$. Let depth of C is denoted by d and size is denoted by d. Since the circuit is in NC^1 we can assume the circuit has fanin 2. Since C has fanin 2 and depth d s can be atmost 2^d. We will show we can convert C into a formula of at most size 2^d and depth d.

We will unveil the circuit from the top and go towards bottom. For any depth level i if for any k many gates there is a common child gate then we make k many copies of that child gate, one for each gate at depth level i and each of that child gate remains at depth $i-1$, so that each child gate has fanout 1. At depth i maximum number of gates is 2^{d-i}. So at max with this process of unveiling at depth i we twice the gates in C. So at each depth we at most twice the gates. After the process So at max the size will be after the process is $O(2^{O(d)}s)$. Since $d = O(\log n)$ so size becomes $O(n^{O(1)}s) = poly(n)$. Hence in NC^1 circuits and formulas are equivalent.

□

Problem 2

Prove that $TC^0 \subseteq NC^1$ using Redundant Algebra

Solution: Let us denote the addition of two n bit numbers using redundant algebra is denoted by $ADDR$. Now we have showed that $ADDR \in NC^0$. Now we will first reduce addition of 3 n-bit numbers to addition of 2, $(n+1)$-bit numbers. Let a, b, c are 3, n-bit numbers. where $a = a_{n-1} \dots a_1 a_0$, $b = b_{n-1} \dots b_1 b_0$ and $c = c_{n-1} \dots c_1 c_0$ where each $a_i, b_j, c_k \in \{0, 1\}$. Nowwe can construct two numbers x and y each having $(n+1)$-bits where

$$x_i = a_i \oplus b_i \oplus c_i \text{ and } y_i = (a_{i-1} \wedge b_{i-1}) \vee (b_{i-1} \wedge c_{i-1}) \vee (c_{i-1} \wedge a_{i-1}) \quad \forall\, 0 \leq i \leq n-1,\ y_0 = 0$$

Now $x + y = a + b + c$. So we have reduced the addition of3, n-bit numbers to addition of 2, $(n+1)$-bit numbers, which is in NC^0 using redundant algebra. Hence

In $ITERADD_{n,n}$ we first

□