CS5110: Complexity Theory

Shreyas Havaldar
CS18BTECH11042

November 17, 2020

1 Exercise Set 14, Lecture 19

1.1 Depth of Circuits

Note: I'm calculating depth as the length of the longest path between the output and the input. That is for a simple AND between 2 input variables, the depth would be 1.

1.1.1 Brute-Force

Depth = 2, evident from the circuit.

1.1.2 Divide-And-Conquer

Depth = $O(\log n)$, as for each level of the circuit we divide the input variables by half, so we need to find the number of levels we have until we reach a level where there are only 4 inputs, thus n=4, is our base case for recursion. We know the depth is 3 for n=4 and now we solve the recursion using this base case. The recursion we solve would be $f(n) = f\left(\frac{n}{2}\right) + c$. The constant c refers to the path from input to the threshold recursion and then from threshold to output gate.

1.1.3 Hiding

Depth = 2, as evident from the circuit.

1.1.4 Partition-and-Conquer

Depth = 3, as evident from the circuit.

1.1.5 Grid

Depth = $O(\log \log n)$. At each level the threshold has input variables reduced to square-root of the upper level, we are basically just solving the recursion for the number of levels until we reach a state with only 4 inputs, thus n=4, is our base case for recursion. We know the depth is 3 for n=4 and now we solve the recursion using this base case. The recursion we solve would be $f(n) = f(\sqrt{n}) + c$. The

constant c refers to the path from input to the threshold recursion and then from threshold to output gate.

1.1.6 Binary-Partition

Depth = 3, as evident from the circuit drawn in the previous exercise.

1.2 NC^0

Class of all poly size circuits, with poly wires and bounded fan-in with constant $O(1)$ depth.

It is pretty obvious that any function that is dependent on non-constant number of input bits, that is values of all its n inputs would not be in NC^0. As all the n inputs would be the inputs to some gate and thus we would not be able to have either a bounded fan-in or an constant depth. Depending on our construction of the circuit, either the depth would depend on n, or the fan-in and thus such a function will not belong to NC^0. Let the depth of the circuit be denoted by d, and fan-in by w. We would only be able to have w^d inputs in total to all our gates, which is a constant since both fan-in and depth is constant, thus we would not be able to compute such a function correctly for $n > w^d$.

One simple example of a function not in NC^0 would be the AND of all n input bits, say $f = (x_1 \wedge x_2 \ldots \wedge x_n)$, where x_i denotes the i^{th} input bit.

1.3 $AC^0 \subseteq NC_1$

To convert a constant depth unbounded fan-in circuit to a $O(\log n)$ depth bounded fan-in circuit.

So we need to convert a gate with unbounded fan-in to possibly multiple gates with bounded fan-in but non-constant depth. As we know a gate has the basis of $\{\vee, \wedge, \neg\}$, for a gate with n input bits we simply convert this gate wlog say with the operator $\wedge$ to a tree of depth $\log n$ with $(\frac{n}{2} + \frac{n}{2} + \frac{n}{4} + \frac{n}{8} + \frac{n}{16} \ldots + 1)$ gates all with a operator of $\wedge$. At the lowermost level we would have $\frac{n}{2}$ gates each with two input bits to it serving as input. Then at one level higher we would have $\frac{n}{4}$ gates with 2 gates from the lower level serving as inputs to each of these gates and so on until we reach the top-most level, the root of the tree in some sense where there is only one gate which would give us the output taking the two inputs which come to it from the lower level. Can intuitively think of converting an n-ary tree of depth = 1, to a binary tree of depth = $\log n$, both of which are equivalent logically, that is give the same output for the same input variables.

Thus we can perform this operation for every gate with unbounded fan-in and achieve a constant fan-in for all gates but with the circuit depth magnified to $O(\log n)$. And thus we can convert every circuit belonging to the class AC^0 to an equivalent circuit in class NC^1.

Hence Proved!