IITM-CS6840: Advanced Complexity Theory Mar 13, 2012

Lecture 38 : Circuit Lower Bound Problem

Lecturer: Jayalal Sarma *Scribe: Rahul CS*

THEME: In this lecture, we approach the question whether $\mathsf{NP} \subseteq \mathsf{P}/Poly$ in the context of boolean circuits. We also discuss some trivial size and depth lower bounds and Upper Bounds of simple functions like PARITY, ADD, their circuits. The class NC.

We use the fact that $\mathsf{P}/poly = PSIZE$. Now the question $\mathsf{NP} \subseteq \mathsf{P}/poly$ could be rephrased as asking whether there exists a polynomial size circuit for the $\mathsf{NP} - complete$ problem SAT. Any superpolynomial lower bound for boolean functions in NP will prove $\mathsf{NP} \not\subseteq \mathsf{P}/poly$.

We still cannot prove super-linear lower bounds for circuits with $\{\wedge, \vee and \neg\}$ as basis. Shannon showed that most of the boolean functions require $\Omega(\frac{2^n}{n})$ size. His arguments were counting based. It does not characterize the circuit family. Count how many different boolean functions of n variables can be computed using a given number of elementary operations, and compare this number with the total number 2^{2^n} of all boolean functions.

Later Lupanov came up with an upperbound saying that every Boolean function can be implemented using $(1 + O(1))\frac{2^n}{n}$ gates.

If $f|_{x_i=0} \neq f|x_i = 1$ we say that function is sensitive to every input(or the function is said to be nondegenerated). In such circuits, we have trivial lower bound of $S \geq n - 1$ on size and $d = \lceil \log n \rceil$ on depth.

Suppose C be a minimal size circuit computing function f. Each gate has one outgoing edge n input edges hence there will be $s - 1 + n$ edges(final output won't be counted) in total. Each gate has atmost two incoming edges which means, $no\ of\ edges \leq 2s$. $s + (n-1) \leq 2s$ $\Rightarrow s \geq n - 1$.

Open question: Is there a boolean function $\{f_n\}_{n\geq 0}$ that require $\omega(n^{1+\epsilon})$ size for any circuit computing this function.

Another puzzle topic is, sorting has $O(n \log n)$ super linear algorithm. Do we have an $O(n \log n)$ lower bound for circuits that does sorting? Ajtai and J. Komlos has come up with $O(n \log n)$ sorting network.

Consider the following function $Parity(x_1, x_2, ..., x_n)$,

$$Parity(x_1, x_2, ..., x_n) = \begin{cases} 1 & \text{if } (\Sigma x_i \bmod 2) = 1 \\ 0 & \text{otherwise} \end{cases}$$

With $\oplus$ gates having fanin two, Parity can be implemented using a circuit of depth $\log n$ and

number of gates $n-1$. If we change basis, size of circuit may change, but only by a constant factor. This is because of the fact that, each element in one basis can be implemented by a circuit for that element composed of elements from the other basis and vice versa.

Thus if we replace $\oplus$ gate with minimal gadgets composed of $\{\wedge, \vee, \neg\}$ so that circuit remains to be of depth $O(\log n)$ and size of $O(n)$

Addition of two n-bit numbers

Consider a circuit that computes the function $ADD_n : \{0,1\}^n \times \{0,1\}^n \Rightarrow \{0,1\}^{n+1}$. Let $(a_1, a_2, ...a_n)$ and $(b_1, b_2, ..., b_n)$ be the inputs and $(s_1, s_2, ..., s_{n+1})$ all the binary numbers are indexed from MSB to LSB.

The trivial circuit for addion has depth $O(n)$ since computation of each bit require the carry generated by the previous bit(the less significant bit). If s_i is the i^{th} bit in the sum, s_i depends on a_i, b_i, c_i where c_i is the carry from the addition of previous bits. and a_i and b_i are i_{th} bit in the input respectively.

It could be noticed that the i^{th} carry bit takes the value 1, if it is generated in some previous less significant bit $j > i$ and got carried all the way from j to i. A carry bit get generated at some index if both the inputs at that index are $1(a_i \wedge b_i = 1)$. Similarly carry get propagated from index k to $k+1$ if one of the input bits in the k^{th} index are $1(a_k \vee b_k = 1)$. Thus,

$$c_i = \bigvee_{j=0}^{j>i} [(a_j \wedge b_j) \wedge (\bigwedge_{i<k\leq j} (a_i \vee b_i))]$$

We could reduce the depth of the circuit further by blockwise carry generation and propagation. We divide the total number of bits in to blocks and we deal blocks in the same way as bits. *i.e.* each block either generates carry or propagates carry to higher significant blocks. And we implement the same add function within blocks.

This procedure gives a circuit of $\log n$ depth and size $O(n^3)$.

Complexity Classes

In the following section, we discuss few of the standard complexity classes.

class NC

N in NCstands for Nick Pippenger. This class represents a family of circuits with following characteristics. All the circuits in the family are implemented using gates of bounded fanin.

$$NC^1 = \{L | function\ can\ be\ computed\ by\ polynomial\ size\ O(\log n) depth\ circuits\}$$

For example Addition and Parity are in NC^1 as we have already seen. In general, class NC^i is defined to be,

$$NC^i = \{L | function\ can\ be\ computed\ by\ polynomial\ size\ O(\log^i n) depth\ circuits\}$$

Class $\mathsf{NC}$is defined to be $\mathsf{NC} = \cup_{\{i \geq 0\}} NC^i$. This class in general, is a measure of efficiency of parallel algorithms.

Class CC^0 represents class of circuits that are of constant depth, implemented using mod_m gates.