Topics in Complexity Theory	23 Feb, 2015

Lec. 9: Lower Bounds

Lecturer: Neeraj Kayal *Scribe: Vineet Nair*

In this lecture we will look at some of the lower bound problems we will address to in this course. In a lower bound problem we have some model of computation C (for example Turing macines, boolean circuits, arithmetic circuits etc) computing a family of functions $\{f_1(x_1), f_2(x_1, x_2), ..., f_n(x_1, x_2, ..., x_n)\}$, we want an explicit family of functions τ such that computing τ via C requires large size. In this course we will mostly address lower bounds for arithmetic circuits. Below we define arithmetic circuits.

Definition 9.0.1 (Arithmetic Circuit)**.** *An arithmetic circuit C over the field $\mathbb{F}$ and the set of variables X is defined as a directed acyclic graph. The leaves of the graph (with indegree equal to zero) are the input nodes and are labelled by input variables or field elements. The other gates are labelled by $\times$ or $+$ and are refered to as product gates or sum gates respectively.*

Every gate of indegree zero is called an input gate (even when the gate is labelled by a field element). Every gate of out-degree zero is called an output gate. The edges in an arithemtic circuit are labelled by field elements. If an edge (u, v) is labelled by $\alpha \in \mathbb{F}$ then the input to v is α times the output of u. Similar to monotone boolean circuits we have monotone arithmetic circuits.

Definition 9.0.2 (Monotone Arithmetic Circuit)**.** *An arithmetic circuit C over reals in which all edges and leaves are labelled by non-negative real numbers.*

Basic facts about Arithmetic Circuits: From figure 1 we see that if we have two addition gates (or two multiplication gates) such that one is a child of other, we can group them together to form a single gate.

Similarly figure 2 shows, we can push all the constants on the edges, in the circuit, to edges connected to the leaves. Thus we can assume without loss of generality that all constants on edges except those connected to leaves are 1.

We can do a similar thing for boolean circuits. We can push all the negation gates in the circuit, down to the leaves using De Moivre's theorem as shown in figure 3.

Next we show that pushing all the negation gates in the circuit, down to the leaves increases the size at most by a factor of two. We show this using straight line programs(SLPs). We know there is a bijection between boolean straight line programs and boolean circuits such that the size of straight line program is equal to number of nodes in circuit. Say we have a circuit C, we can reduce it to a straight line program by traversing the circuit one level at a time starting from leaves and for each node covered in the circuit we write an equivalent line in SLP. For example the corresponding straight line program for the circuit shown on the left (before the reduction) in figure 3 is:

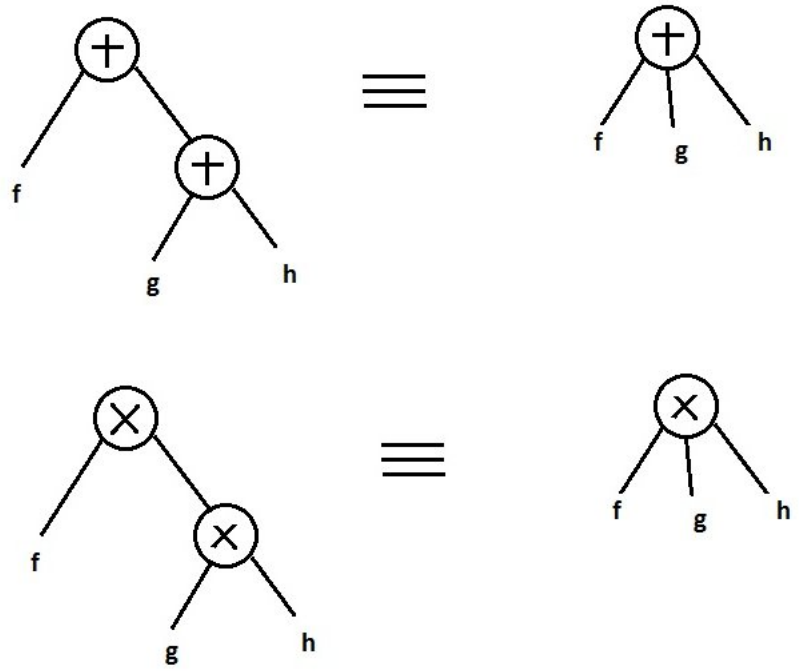

Figure 1: Reduction from two gates to one

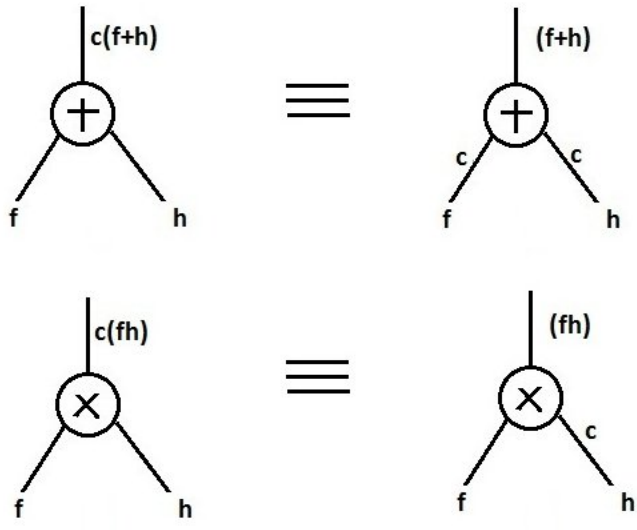

Figure 2:

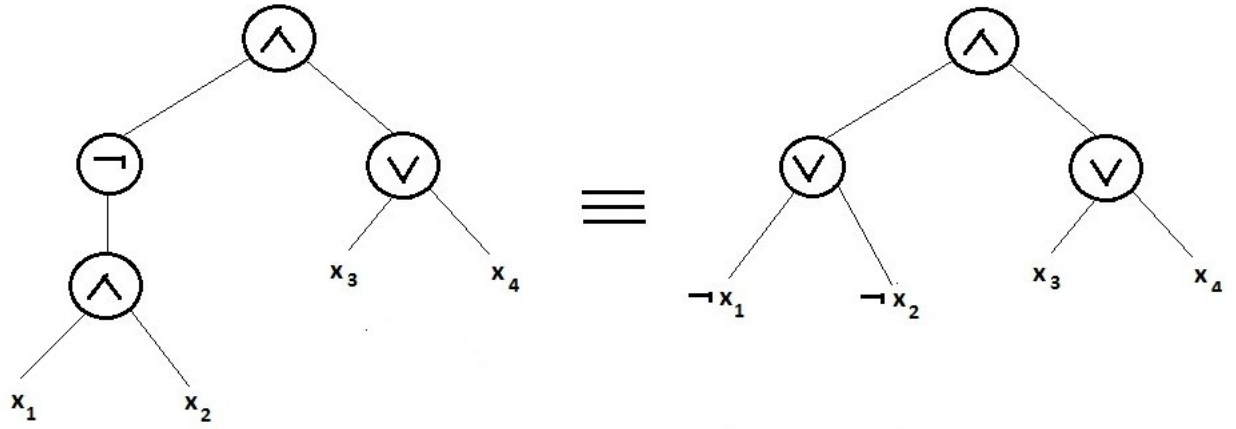

Figure 3:

$$
\begin{aligned}
S_1 &= x_1 \\
S_2 &= x_2 \\
S_3 &= x_3 \\
S_4 &= x_4 \\
S_5 &= S_1 \wedge S_2 \\
S_6 &= S_3 \vee S_4 \\
S_7 &= \neg S_5 \\
S_8 &= S_7 \wedge S_6
\end{aligned}
$$

Once we reduce the given circuit to a straight line program, we compute the negation of each line in the straight line program. For example for the above straight line program we have:

$$
\begin{aligned}
S_1 &= x_1 & T_1 &= \neg S_1 \\
S_2 &= x_2 & T_2 &= \neg S_2 \\
S_3 &= x_3 & T_3 &= \neg S_3 \\
S_4 &= x_4 & T_4 &= \neg S_4 \\
S_5 &= S_1 \wedge S_2 & T_5 &= \neg S_5 \\
S_6 &= S_3 \vee S_4 & T_6 &= \neg S_6 \\
S_7 &= \neg S_5 & T_7 &= \neg S_7 \\
S_8 &= S_7 \wedge S_6 & T_8 &= \neg S_8
\end{aligned}
$$

It is easy to see we can construct a circuit such that all the negation gates are at the leaves from such a straight line program. Since we have just doubled the size of the straight line program by computing the negation of each line in the original straight line program, the size of the circuit we get is at most twice the size of original circuit.

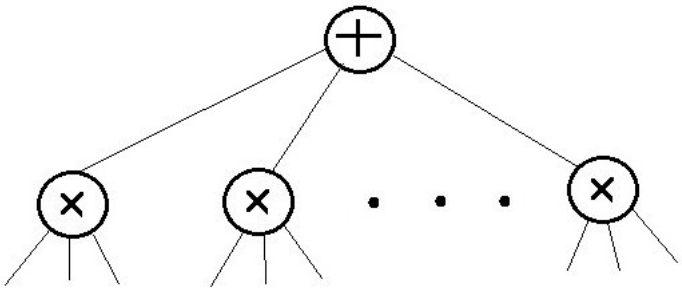

Figure 4: First model of depth two circuit

9.1 Lower bounds on circuits

Jerrum and Snir gave exponential lower bounds for montone arithmetic circuits. They consider the permanent of $n \times n$ matrix. Consider the symbolic $n \times n$ permanent.

$$Perm(A) = \begin{vmatrix} x_{11} & x_{12} & ...x_{1n} \\ x_{21} & x_{22} & ...x_{2n} \\ \cdot & \cdot & ... \quad \cdot \\ \cdot & \cdot & ... \quad \cdot \\ x_{n1} & x_{n2} & ...x_{nn} \end{vmatrix} = \sum_{\sigma:[n]\to[n]} \prod_{i=1}^{n} x_{i\sigma(i)}$$

Since there are $n!$ monomials in permanent we know there exists a trivial monotone circuit of size less than $n!$. Jerrum and Snir showed that this is almost optimal.

Theorem 9.1 ([JS82]). *Every monotone arithmetic circuit computing the permanent has size equal to* $2^{\theta(n)}$

We have an exponential lower bound for boolean montone circuits too by Razborov which was later improved upon by Alon and Boppana. They showed the lower bound for clique. We define the clique problem below and then give their result.

Clique: Given a graph G=(V,E), where $|V| = n$, and an integer k, is there a subset of size k such that $\forall(i,j) \subseteq S$, the edge $(i,j) \in E$.

Theorem 9.2 ([Raz85],[AB87]). *Every monotone boolean circuit for clique has size equal to* $2^{\Omega(n)}$

Before we state more results, we define two different 'depth two arithmetic circuit'. The first model has sum gate as root and product gates at level two as shown in figure 4, whereas the second model has product gate as root and sum gates at level two as shown in figure 5. We understand both models of depth two circuits completely. The size of the first model is equal to the number of monomials in the circuit. Hence any depth two circuit computing the permanent or the determinant requires at least $n!$ size. Next we show that the determinant and permanent are irreducible. This would imply the second model of depth two circuit cannot compute the permanent or determinant polynomials.

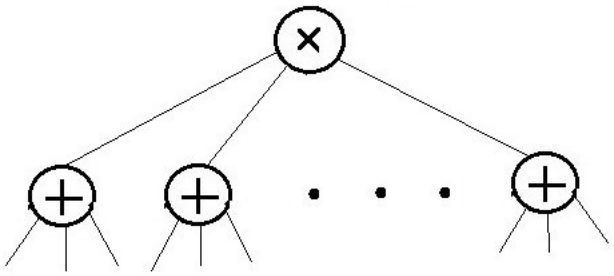

Figure 5: Second model of depth two circuit

Lemma 9.1.1. *Determinant and permanent polynomials are irreducible.*

Proof. We would prove the claim for determinant, the proof for permanent is similar. Consider the symbolic $n \times n$ determinant.

$$det(A) = \begin{vmatrix} x_{11} & x_{12} & ...x_{1n} \\ x_{21} & x_{22} & ...x_{2n} \\ \cdot & \cdot & ... \quad \cdot \\ \cdot & \cdot & ... \quad \cdot \\ x_{n1} & x_{n2} & ...x_{nn} \end{vmatrix} = \sum_{\sigma:[n]\to[n]} sgn(\sigma) \prod_{i=1}^{n} x_{i\sigma(i)}$$

Let $X = \{x_{11}, x_{12}, ..., x_{nn}\}$. Say for contradiction the $det(A) = P_1P_2$ where P_1 and $P_2 \in \mathbb{F}[\mathbb{X}]$. Let $Var(P_i)$ denote the set of variables appearing in P_i. We claim that $Var(P_1) \cap Var(P_2) = \emptyset$. Say for contradiction $Var(P_1) \cap Var(P_2) \neq \emptyset$. This imples there exists a variable say x_{ij} that belongs to both $Var(P_1)$ and $Var(P_2)$. Thus there would exist a monomial in P_1P_2 such that the degree of variable x_{ij} in that monomial is greater than one. Since det(A) is a multilinear polynomial, we get a contradiction.

Consider the variables x_{11} to x_{1n} that appear in the first row of A. Since every monomial in $det(A)$ has n variables each from a unique row and column and $Var(P_1) \cap Var(P_2) = \emptyset$, these variables belong to either $Var(P_1)$ or $Var(P_2)$. W.l.o.g assume they belong to $Var(P_1)$. Similarly we can conclude the variables in the first column of A, $\{x_{11}, x_{21}, ..., x_{nn}\}$ belong to $Var(P_1)$. Hence considering each column at a time we conclude $Var(P_1) = X$, which implies $Var(P_2) = \emptyset$. Thus P_2 is just a field constant, a contradiction. □

Our knowledge of lower bounds at present is very limited. As far as we know:

1. Permanent of an $n \times n$ symbolic matrix could be computed by circuits of size $100n^2 \log n$.

2. Clique could be computed by circuits of size $6n^2$.

3. Permanent could be computed by depth three circuits of size (number of nodes) n^4.

4. Clique could be computed by depth two circuits of size n^4.

We state some of the well known lower bounds below. Grigoriev and Karpinski proved the following lower bound.

Theorem 9.3 ([GK98]). *Any depth three arithmetic circuit computing the determinant of an $n \times n$ symbolic matrix over a fixed finite field must have $2^{\Omega(n)}$ size.*

It is not known whether the above theorem holds over $\mathbb{Q}$. Also it is still open to show an exponential lower bound for depth four circuits over a fixed finite field. The next theorem was also stated in lecture note 4. It shows a lower bound for the 'parity' function.

Theorem 9.4 ([AKS83],[FSS84]). *Any $\triangle$ depth circuit for computing the parity of n bits using AND, OR and NOT gates (of unbounded fan-in) must have size $2^{O(n^{\frac{1}{\triangle}})}$.*

This was later made optimal by Håstad. Using his switching lemma [Hås86], Håstad showed that any constant depth circuit computing the parity function requires exponential size. The next theorem in some sense shows why we are still not able to make progress on lower bounds in the boolean world.

Theorem 9.5 ([RR97]). *If 'pseudorandom functions' exist then a large set of techniques we know (natural proofs) cannot prove superpolynomial lower bounds.*

It is conjectured by most cryptographers and complexity theorists that pseudorandom functions do exist, hence the above theorem implies most techniques used to prove lower bounds in the boolean world may not be strong enough to prove superpolynomial lower bounds. Theorem 9.5 does not apply to arithmetic circuits.

References

[AB87] Noga Alon and Ravi B. Boppana. The monotone circuit complexity of boolean functions. *Combinatorica*, 7(1):1–22, 1987.

[AKS83] Mikls Ajtai, Jnos Komls, and Endre Szemerdi. An 0(n log n) sorting network. In *Proc. 15th Annual ACM Symposium on the Theory of Computing*. ACM, 1983.

[FSS84] Merrick Furst, James B Saxe, and Michael Sipser. Parity, circuits, and the polynomial-time hierarchy. *Math. Syst. Theory*, 1984.

[GK98] Dima Grigoriev and Marek Karpinski. An exponential lower bound for depth 3 arithmetic circuits. In *Proc. 13th Annual ACM Symposium on the Theory of Computing*, pages 577–582, 1998.

[Hås86] J. Håstad. Almost otimal lower bounds for small depth circuits. In *Proc. 18th Annual ACM Symposium on the Theory of Computing*. ACM, 1986.

[JS82] Mark Jerrum and Marc Snir. Some exact complexity results for straight-line computations over semirings. *J. ACM*, 29(3):874–897, 1982.

[Raz85] Alexander A. Razborov. Lower bounds on the monotone complexity of some boolean functions. *Mathematics of the USSR, Doklady*, 31:354–357, 1985.

[RR97] Alexander A. Razborov and Steven Rudich. Natural proofs. *J. Comput. Syst. Sci.*, 55(1):24–35, 1997.