,12

CS277: Concrete Complexity **Fall 1996**

Lecture 11,12: October 1, 3, 1996

Lecturer: Uri Zwick *Scribe: Ashwin Nayak*

11.1 Polynomial size monotone formulae for majority

In this lecture, we will construct polynomial size formulae for majority over the monotone basis $M_2 = \{\vee, \wedge\}$. We will actually construct $O(\log n)$ depth monotone circuits, which would clearly give us polynomial size monotone formulae. We had seen that the existence of a polynomial sized formula for a function f over B_2 implied existence of an $O(\log n)$ depth circuit, and hence a formula with the same depth, over B_2. A similar result holds in the monotone basis case.

Exercise 11.1 *Prove that* $D_{M_2}(f) = \Theta(\log L_{M_2}(f))$.

The existence of an $O(\log n)$ depth monotone formula for majority follows from a theorem of Ajtai, Komlos and Szemeredi about sorting networks.

11.1.1 A construction from sorting networks

Definition 11.1 *A* sorting network *is a circuit composed of comparator units (see Figure 11.1) which sorts its input numbers. Any input or output of a comparator in the network can feed only one other comparator.*

Example: Figure 11.3 shows a network that sorts any given four numbers. ■

The comparators in a sorting network can be arranged in levels given by their topological ordering. The number of levels in the network is called its depth. The size of a network is the number of comparators in it.

Theorem 11.2 (Ajtai, Komlos, Szemeredi '83) *There are sorting networks of depth* $O(\log n)$.

The size of the AKS networks is clearly $O(n \log n)$, since each level of the network can accommodate at most $n/2$ comparators.

Since a sorting network correctly sorts all inputs, it can also sort 0-1 inputs. Interestingly, for a network to be able to sort all inputs correctly, it is sufficient that it sorts all 0-1 inputs!

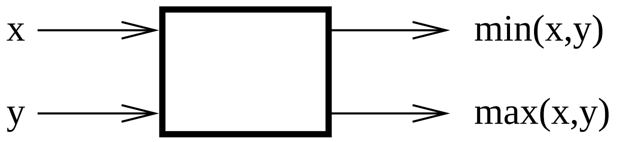

Figure 11.1: A comparator unit. It is usually drawn as in Figure 11.2.

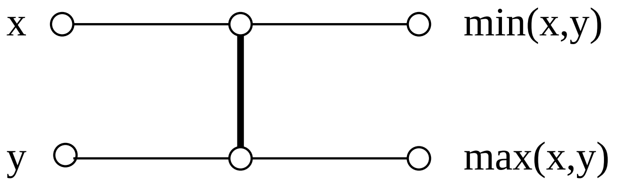

Figure 11.2: Short hand notation for a comparator unit.

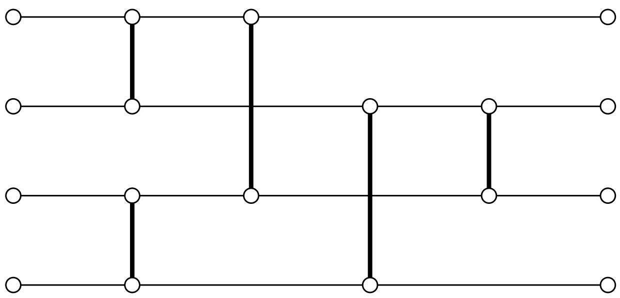

Figure 11.3: A four input sorting network.

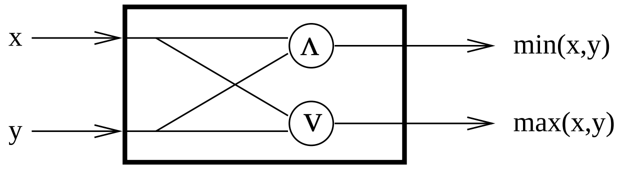

Figure 11.4: An implementation of a 0-1 comparator with M_2 gates.

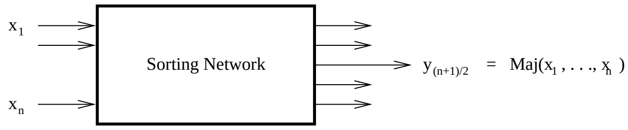

Figure 11.5: A circuit for majority obtained by replacing comparator units in a sorting network by the $\wedge, \vee$ combination shown in Figure 11.4.

Exercise 11.2 (0-1 principle of sorting networks) *If a comparator network sorts all* 0-1 *inputs, then it sorts all inputs.*

Assuming n is odd, the $(n+1)/2$th element in the sorted input 0-1 sequence would correspond to the majority of the input sequence. So if we could implement comparators with M_2 functions, we would get a depth $O(\log n)$ monotone circuit for majority by replacing comparators by their M_2 implementations. And, indeed, we can construct such comparators and majority formulae (see Figure 11.4 and Figure 11.5).

So we have the following corollary to the AKS theorem.

Corollary 11.3 *There exist $O(\log n)$ depth, $O(n \log n)$ size monotone circuits for majority.*

This construction is very unsatisfactory because the big-oh notation in the AKS theorem hides a huge constant, so that the formulae obtained from sorting networks would be much too large. A non-constructive theorem due to Valiant shows the existence of $O(\log n)$ depth circuits for majority which gets rid of this malady. We turn to this next.

11.1.2 Valiant's Method

Theorem 11.4 (Valiant '84) $D_{M_2}(f) \leq 5.3 \log_2 n$.

In more detail, Valiant's result goes like so: Take a complete binary tree of depth $5.3 \log_2 n$ (this depth can be assumed to be even). Label the nodes in the tree level by level alternatingly with $\vee$'s and $\wedge$'s as shown in Figure 11.6.

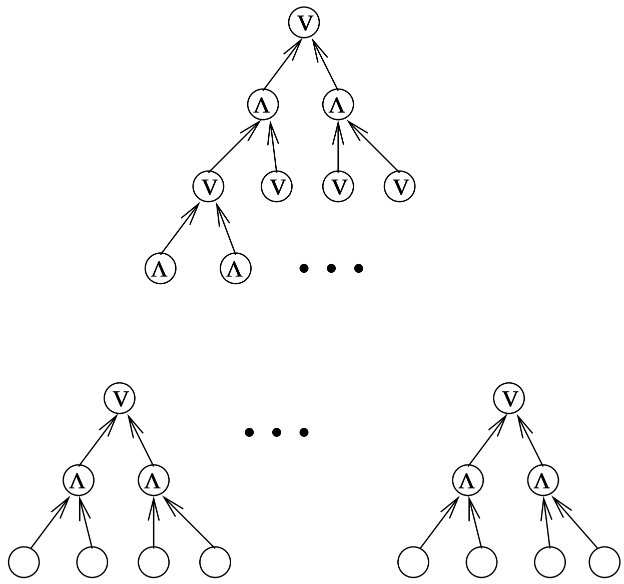

Figure 11.6: An alternating tree of $\vee$'s and $\wedge$'s.

Label each leaf in the following manner: with probability $2\psi - 1$, where $\psi = (\sqrt{5}-1)/2$, label the leaf with the constant '1.' With the complementary probability, $2(1-\psi)$, label the leaf with a random variable from $\{x_1, x_2, \ldots, x_n\}$.

With high probability, the resulting formula computes the majority function!

We need a definition before we proceed to the proof.

Definition 11.5 *Let $f(x_1, \ldots, x_n)$ be a Boolean function. The* amplification function*, $A_f(p)$, of the function f is defined as*

$$A_f(p) = \Pr_{x_1,\ldots,x_n \in_R \{0,1\}} \left[f(x_1, \ldots, x_n) = 1 \right].$$

Let $A_k(p)$ denote the amplification function of an alternating tree with $2k$ levels as used in Valiant's construction.

Lemma 11.6 *For all k, $A_k(p)$ is a monotonically increasing function with fixed points $0, \psi$ and 1. If $2k \geq 5.3 \log_2 n$ then $A_k\left(\psi + \frac{1-\psi}{n}\right) > 1 - 2^{-n}$, and $A_k\left(\psi - \frac{1-\psi}{n}\right) < 2^{-n}$.*

The function $A_k(p)$ has the same shape as $A_1(p)$, which is plotted in Figure 11.7, but asymptotically tends to the unit step function with the transition at $p = \psi$ as k increases.

We will first show how we prove Valiant's theorem using this lemma.

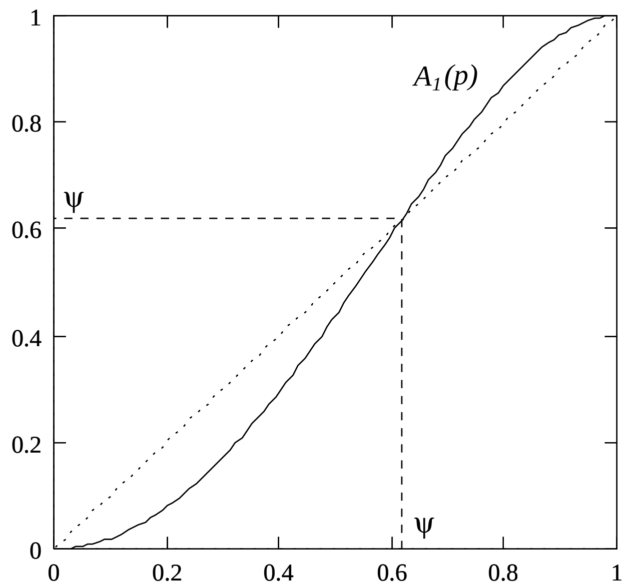

Figure 11.7: $A_1(p)$. The function has three fixed points: $0, 1, \psi = 0.61803\ldots$.

Proof: (of Theorem 11.4) Assume n to be odd. We show that the with positive probability, the formula constructed above outputs the majority of the n inputs. Thus, one of the formulae constructed must compute majority correctly.

What is the probability that the formula outputs the wrong value for a specific input $x = (x_1, \ldots, x_n)$?

Assume that $\mathrm{Maj}(x) = 1$ (the other case is symmetric). The number of 1's in x is at least $(n+1)/2$. So the probability that a leaf gets a 1 is at least

$$(2\psi - 1) + 2(1-\psi)\frac{n+1}{2n} = \psi + \frac{1-\psi}{n}.$$

Therefore the probability that the formula outputs 0 is at most

$$1 - A_k\left(\psi + \frac{1-\psi}{n}\right) < 2^{-n}.$$

Thus, from the union bound, the probability that the formula is wrong on any input is less than 1. ∎

It only remains to show that the lemma is true.

Proof: (of Lemma 11.6)

We will prove the lemma in two stages. In the first stage, we argue that in $k = 1.64 \log_2 n$ levels, the probabilities look like so:

$$\begin{aligned} A_k\left(\psi - \frac{1-\psi}{n}\right) &< 1/4 \\ A_k\left(\psi + \frac{1-\psi}{n}\right) &> 3/4. \end{aligned}$$

In the second stage, we will show that we can get the required probabilities in a further $O(\log n)$ levels by showing that for $k = \log_2 n$,

$$\begin{aligned} A_k(1/4) &< 2^{-n} \\ A_k(3/4) &> 1 - 2^{-n}. \end{aligned}$$

We will actually show only the first two inequalities. The other two follow from an analogous piece of reasoning which is omitted from the proof.

Observe that $A_k(p)$ satisfies the following properties:

$$\begin{aligned} A_1(p) &= 2p^2 - p^4 \\ A_1'(p) &= 4(p - p^3) \\ A_k(p) &= A_1(A_{k-1}(p)) \\ A_k'(p) &= A_1'(A_{k-1}(p))A_{k-1}'(p) \end{aligned}$$

Here, the notation f' is used to denote the derivative of the function f.

The first stage. Note that $A_1'(p)$ is a monotonically decreasing function for $p \geq \phi$ where $\phi = 1/\sqrt{3}$ is point where $A_1'(p)$ attains its maximum (see Figure 11.8). Also note that $A_1'(\psi) = (2\psi)^2$.

Claim 11.7 *If $\epsilon(2\psi)^{2k} \leq \psi - \phi$, then $A_k(\psi - \epsilon) \leq \psi - (2\psi)^{2k}\epsilon$.*

Proof: The proof is by induction. For $k = 0$, equality holds. For $k > 1$,

$$\begin{aligned} A_k(\psi - \epsilon) &= A_1(A_{k-1}(\psi - \epsilon)) \\ &\leq A_1(\psi - \epsilon(2\psi)^{2k-2}) \\ &= A_1(\psi) - \epsilon(2\psi)^{2k-2}A_1'(\theta) \quad (\theta \in [\phi, \psi]) \\ &\leq \psi - \epsilon(2\psi)^{2k} \end{aligned}$$

The first inequality follows from the induction hypothesis, the next equality from the mean value theorem, and the final inequality follows from the monotonically decreasing nature of $A_1'(p)$ for $p > \phi$. ∎

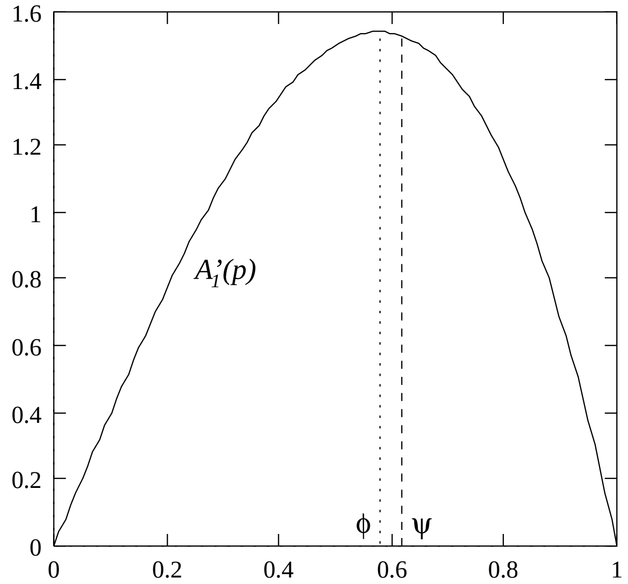

Figure 11.8: The function $A_1'(p)$. It attains its maximum at $\phi = 1/\sqrt{3} < \psi$.

Setting $\epsilon = \frac{1-\psi}{n}$ and $2k = \log_{2\psi} \frac{\psi-\phi}{\epsilon} \leq 3.28 \log_2 n$, we get $A_k\left(\psi - \frac{1-\psi}{n}\right) \leq \phi$. Using the fact that $A_k(\phi)$ tends to 0 as k tends to ∞ (since 0 is a stable fixed point of $A_1(p)$), we can add a constant number more of alternating layers on top of the above tree to push $A_k(\psi - \epsilon)$ below $1/4$.

Now the second stage.

Claim 11.8 *$A_k(p) < \frac{1}{2}(2p)^{2^k}$ for $p > 0$.*

Proof: The proof is by induction. For $k = 1$, we see that $A_1(p) = 2p^2 - p^4 < 2p^2$. For $k > 1$,

$$\begin{aligned} A_k(p) &= A_1(A_{k-1}(p)) \\ &< A_1\left(\frac{1}{2}(2p)^{2^{k-1}}\right) \\ &< 2\left(\frac{1}{2}(2p)^{2^{k-1}}\right)^2 \\ &= \frac{1}{2}(2p)^{2^k} \end{aligned}$$

■

Therefore, $A_k(\frac{1}{4}) < (\frac{1}{2})^{2^k}$, and so with $k = \log_2 n$, $A_k(\frac{1}{4}) < 2^{-n}$, which completes the proof.
■

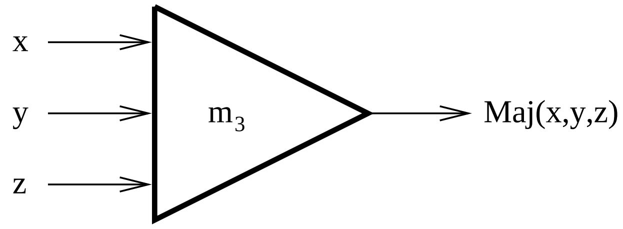

Figure 11.9: Majority gate over three inputs.

Exercise 11.3 *Construct majority formulae over the basis $\{m_3\}$, where m_3 is the three input majority gate shown in Figure 11.9.* Note: One can easily express the basis M_2 in terms of m_3 gates, and hence use Valiant's construction to get the required circuit. We are interested in a *direct* construction.

Exercise 11.4 *Construct an approximate majority formula over U_2. If the number of 1's is less than $0.49n$, the formula should output a zero. If the number of 1's is greater than $0.51n$, then the formula should output a one. Otherwise, we do not care about the output.*

11.2 Non-deterministic and probabilistic circuits

Definition 11.9 *A* non-deterministic circuit *for a Boolean function, $f(x_1, \ldots, x_n)$, is a circuit C with inputs $x_1, \ldots, x_n$, and auxiliary inputs $y_1, \ldots, y_m$, where $m = \text{poly}(n)$, such that if $f(x_1, \ldots, x_n) = 1$, then there is a setting for the inputs y such that $C(x_1, \ldots, x_n, y_1, \ldots, y_m) = 1$, and if $f(x_1, \ldots, x_n) = 0$, then for no value of the variables y is the output of the circuit equal to 1.*

Definition 11.10 *A* probabilistic circuit *for a Boolean function $f(x_1, \ldots, x_n)$ is a circuit with auxiliary inputs y (as above) such that the the following holds:*

$$\Pr_{y_1,\ldots,y_m \in_R \{0,1\}}[C(x_1, \ldots, x_n, y_1, \ldots, y_m) = f(x_1, \ldots, x_n)] > 2/3.$$

I.e., the probability (over random choices of the auxiliary variables) that the circuit computes the function correctly is greater than 2/3.

In the non-uniform setting, probabilistic ciruits are no more powerful than ordinary "deterministic" circuits!

11.3 Solution to exercises

Solution: (of Exercise 4) We need to be able to simulate an NL machine with an $O(\log^2 n)$ depth circuit of polynomial size. That this is possible follows from the observations given below:

- We can construct a graph over the instantaneous descriptions (IDs) of the NL machine with directed edges from one ID to the another if and only if there is a one step transition of the machine which transforms the first to the second. In this graph, there would be a path from the initital ID to one of the final configurations if and only if the machine has an accepting computation on the given input. The number of distinct IDs for the computation would be polynomial in the input size because the space used is only logarithmic and we are considering the IDs for a fixed input.
- The existence of a path from some vertex s to another vertex t in a graph with n vertices and adjacency matrix A can be detected by raising A to the nth power, and checking the (s,t)th entry in the resulting matrix. (That the resulting matrix actually gives the transitive closure of the graph can be verified easily.)
- The adjacency matrix of the graph of IDs can be computed in constant depth, since it involves just checking local regions of the ID with the help of the constant sized description of the machine.
- Multiplying two $n \times n$ Boolean matrices takes $O(\log n)$ depth and size $O(n^3)$. Raising it to the n power takes $O(\log n)$ matrix multiplications using the squaring trick for exponentiation. So computing A^n takes $O(\log^2 n)$ depth and $O(n^3 \log n)$ size for an $n \times n$ matrix A.

Note that we have omitted to check whether this construction is logspace uniform. This verification is left to the reader. ∎