Recent advances in Complexity CIS 6930/CIS 4930 November 7, 2002

Lecture 16

Lecturer: Dr. Meera Sitharam Scribe: Erwin Jansen

1 Introduction

Today we continued with the proof of a lower bound for the monotone circuit depth of the boolean function *stcon*. We want to show a depth lower bound for *stcon* by using the following:

- $D(R_{stcon}) = d(stcon)$, (and $D(M_{stcon}) = d_m(stcon)$). i.e. the (monotone) circuit depth of *stcon* is equivalent to the protocol depth of R_stcon (M_stcon).
- A communication protocol for M_{stcon} can be converted to one for R_{fork} with the same depth.
- We establish a lower bound for $D(R_{fork})$. Particulary we will show a lower bound of $\Omega(\log l \log w)$. This will transfer to a lower bound for $D(M_{stcon})$ of $\log^2 n$ if we take $w = l = n$.

First we recall,
FACT 1.
(Zia proved) Circuit depth is equal to the communication complexity.

$$d(f) = D(R_f)$$

FACT 2. Monotone circuit depth is equal to the corresponding monotone communication depth.

$$d_m(f) = D(M_f)$$

Zia didn't prove this but the proof is exactly the same.

Exercise

Exercise 1 *Show that* $D(R_{fork}) \leq \Theta(\log l \log w)$

Exercise

Exercise 2 *Show that this implies a* formula *size*[1] *lower bound for stcon of* $\Omega(n^{\log n}) = 2^{\log^2 n}$.

Now we will continue by investigating the lower bound for communication complexity of R_{fork}. Let us first give a definition of a protocol that is (α, l)-correct for R_{fork}:

(α, l) protocol

Definition 1 *A (α, l) approximation protocol P for R_{fork} on strings of length l is an protocol that is correct on some subset $S \subseteq X \cup Y$ such that if $x, y \in S$, then P is correct and $|S|$ is at least an α fraction of $X \cup Y$.*

We are going to use this definition to lower the depth (the amount of bits communicated) of the protocol. By trading depth for accuracy we can eventually show that a correct protocol must have a certain a minimum depth. In order to use this proof strategy we will introduce two lemmas. One lemma tells us that we can sacrifice depth for accuracy. Potentially the depth lower bound strategy would be to assume to the contrary that there is a small depth circuit computing the function, apply this lemma repeatedly thereby decreasing depth of the circuit down to 0, while still reasonably well approximating the function, which would result in a contradiction.

However the lemma results in a decay of accuracy at such a high rate that no useful approximation of the function is possible once the depth gets down to 0. Hence this strategy alone is not enough.

Now, we use another lemma, which was quite novel at the time of discovery and quite different from the lower bound techniques seen so far. This lemma tells us that we can increase or amplify the accuracy if we decrease the size of the inputs (or decrease the size of the domain in a particular way).

Lemma 1 *A (α, l) protocol P of depth c for R_{fork} can be converted into a $(\frac{\alpha}{2}, l)$-protocol P' of depth $(c-1)$*

Notice that we can interpret α as a parameter that defines the size of the subset of our domain that will still give a correct output.

Exercise

Exercise 3 *Proof this lemma. Assume w.l.o.g. that Alice sends the first bit in P. Now for P' Alice doesn't send this bit, but Bob & Alice will assume that this bit is something (what?) and proceed using $(c-1)$ bits. This guarantees that P' is correct on $\frac{\alpha}{2}$ factors (why?).*

The other lemma is a key lemma, novel at the time of discovery, which is also called the accuracy amplification lemma: it tells us we can decrease the chance of mistakes when we decrease the size of our input:

Lemma 2 (Amplification Lemma) *If $\alpha \geq \frac{\lambda}{w}$ for a large enough constant λ then a (α, l) protocol P of depth c-bits can be converted to a $(\frac{\sqrt{\alpha}}{2}, \lfloor\frac{l}{2}\rfloor)$ protocol of the same depth.*

Note: Lemma 1, Lemma 2 and the definition of (α, l) protocol can be stated directly in terms of monotone circuit depth for st-con.

The proof uses the following claim:

Claim 3 *Consider an $n \times n$ 0-1 matrix. Let m be the number of 1s in it, and m_i be the number of 1s in the i-th row. Denote by $\alpha = m/n_2$ the fraction of 1-entries in the matrix and by $\alpha_i = m_i/n$ the fraction of the 1-entries in the i-the row. Then either (a) there is some row i with $\alpha_i \geq \sqrt{\alpha/2}$ or (b) the number of rows for that $\alpha_i \geq \alpha/2$ is at least $\sqrt{\alpha/2} \cdot n$.*

[1] A circuit whose underlying graph is a tree

Proof: (Of Claim) Intuitively, the claim says that either one of the rows is "very dense" or there are a lot of rows that are "pretty dense." Consider $\sum_{i=1}^{n} \alpha_i$. On one hand, $\sum_{i=1}^{n} \alpha_i = \sum_{i=1}^{n} m_i/n = m/n = \alpha \cdot n$. On the other hand, suppose both (a) and (b) do not hold. This means that for all rows $\alpha_i < \sqrt{\alpha/2}$ and that for less than $\sqrt{\alpha/2} \cdot n$ rows $\alpha \geq \alpha/2$. Therefore,

$$\sum_{i=1}^{n} \alpha_i < (\sqrt{\alpha/2} \cdot n) \cdot \sqrt{\alpha/2} + n \cdot \alpha/2 = \alpha n.$$

A contradiction. ■

Proof: (of Lemma 2) Let S be the set corresponding to the (α, l) protocol. Consider a matrix whose rows and columns correspond to string s in $\Sigma^{l/2}$ and whose (u, v) entry contains 1 if the string $u \circ v$ is in S and 0 otherwise. Note that by the assumptions on S the density of 1s in the matrix is at least α. Applying the claim to this matrix, we get that it satisfies either (a) or (b). For each of the two cases we construct the desired c-bit $(\sqrt{\alpha}/2, l/2)$ protocol. In case (a) there exists a row, corresponding to some string u, whose density is at least $\sqrt{\alpha}/2$. The new protocol works as follows: on input $x, y \in \Sigma^{l/2}$ Alice and Bob use the original c-bit protocol on the length- l string $u \circ x$ and $u \circ y$ (and subtract $l/2$ from the output). Because the same string u is concatenated to both x and y, then the output of the protocol is guaranteed to be in the second half of the string. The protocol succeeds whenever the entries corresponding to x and y (in row u) contain 1. The fraction of strings with this property is at least $\sqrt{\alpha/2} > \sqrt{\alpha}/2$, as needed.

In case (b) we need to do something else: Let S' be the set of all rows with density at least $\alpha/2$. We will find two function $f, g : \Sigma^{l/2} \to \Sigma^{l/2}$ and a set S'' and a set $S'' \subseteq S'$ such that the following properties hold:

1. for all $x \in S'', x \circ f(x) \in S$,

2. for all $y \in S'', y \circ g(y) \in S$,

3. for all $x, y \in S'', the strings$f(x) and $g(y)$ are different in all coordinates, and

4. S'' contains $\sqrt{\alpha}/2$, of the strings in $\Sigma^{l/2}$.

Assuming that such functions exist, the new protocol works as follows: on input $x, y \in \Sigma^{l/2}$ Alice and Bob use the original c-bit protocol on the length-l strings $x \circ f(x)$ and $y \circ g(y)$ (each player can modify its own input). By property (3), for all x and y in S'' the output of the protocol is guaranteed to be in the first half of the string, and therefore the protocol succeeds. By property (4) (contained with (1) and (2)), this is a $(\sqrt{(\alpha/2}, l/2)$ protocol.

It remains to prove the existence of such f, g, and S''. Consider $l/2$ subsets A_i of Σ where each A_i is of size $w/2$. If we guarantee that $f(x)$ is a string in $A = A_1 \times \cdots \times A_{l/2}$ and $g(y)$ is a string in $B = \bar{A}_1 \times \cdots \times \bar{(}A)_{l/2}$, then property (3) immediately holds. The idea is to choose each of the A_is at random and to show that this happens with non-zero probability. To simplify the analysis we choose the A_is as follows: We first choose at random $w/2$ strings $v^1, \cdots, v^{w/2}$ each of length $l/2$. Then we define A_i to include the i-th letter in each of these $w/2$ strings and extend it into a set of size $w/2$ randomly. (Note that this indeed gives random and independent A_is.) Now, fix $x \in S'$. We wish to compute the probability that it has an extension $f(x) \in A$ such that $x \circ f(x) \in S$. It is enough to show that with high probability one of the vectors v_j is such an extension. This is because the probability that none of the vectors is good is smaller than $(1-\alpha/2)^{w/2} < e^{-\alpha w/4}$. Therefore, the probability that either A or the corresponding B (that also consists of $l/2$ sets each of size $w/2$) are not good is at most $2e^{-\alpha w/4}$. In other words, for every $x \in S'$ a fraction of $1 - 2e^{-\alpha w/4}$ of the partitions (A, B) is good. Hence, there is a partition that is good for $1 - 2e^{-\alpha w/4}$ of the elements of S'. Let S'' be this set of elements. The fraction of elements in S'' is $1 - 2e^{-\alpha w/4} \cdot \sqrt{(\alpha/2)}$, which is at least $\sqrt{\alpha}/2$, as long as $\alpha \geq \lambda/w$ (for some constant λ). ■

Theorem 4 *Depth of any protocol for R_{fork}*

$$D(R_{fork}) \geq \Omega(\log l \log w)$$

Proof: First notice that $D(R_{fork}) = D(1, l)$. Surely $D(1, l) \geq D(\alpha, l)$. Suppose we take $\alpha = \frac{1}{2w^{1/3}}$. First we apply lemma 1 $\log w$ times, resulting in $\alpha = \frac{1}{2^{\log w} w^{(1/3)}}$. From this we conclude that the depth: $D_{(\frac{1}{2w^{(1/3)}}, l)}(R_{fork}) \geq D_{(\frac{1}{w^{(2/3)}}, l)}(R_{fork}) + \Omega(\log w)$ now we apply lemma 2: $D(\frac{1}{w^{(2/3)}}, l) \geq D(\frac{1}{2w^{(1/3)}}, \frac{l}{2})$. Then we repeat the procedure. Apply lemma 1 $\log w$ times, and then apply lemma 2. Lemma 1 is in the inter loop iterating $logw$ times and Lemma 2 is in the outer loop iterating $logl$ times. ■

Note: While this proof uses communication complexity as a tool, in fact, we could do the whole thing directly using circuit depth. It is clear that the notion of (α, l) correctness etc.. holds equally well for circuits as for protocols. So today's results for R_{fork} could directly be stated as a lower bound on circuit depth for R_{fork}. In fact, stated thus, particularly the accurac

The only result where some work needs to be done to state directly in terms of circuit depth is the (previous lecture's) conversion of the communication protocol of M_{stcon} into one for R_{fork}. To state or "decode" this reduction into a conversion of a circuit for *stcon* into one for R_{fork}, we have to use the original result (done in Zia's lecture) for the conversion of a communication protocol into a circuit.