14. Direct Sum (Part 1) - Introduction

Lecturer: Prahladh Harsha *Scribe: Abhishek Dang*

14.1 Introduction

The "Direct Sum" problem asks how the difficulty in doing multiple instances of a task scales. As witnessed in the following example it is not clear that n times a problem requires n times the resource.

Example - Circuit complexity:
Task f_A: given x, compute $A.x$
A counting argument tells us that $\exists A$ s.t. f_A requires an $\Omega(n^2/\log n)$ sized circuit.
Task $f_A^{(n)}$: given $(x_1, \ldots, x_n)$, compute $(Ax_1, \ldots, Ax_n) = AX$
and the last problem being an instance of matrix multiplication, is known to be solvable with a circuit of size $O(n^{2.38}) \ll n.(n/\log n)$!

14.2 Communication Complexity

For f, g two boolean functions, we ask the question

$$\mathrm{D}(f) + \mathrm{D}(g) \stackrel{?}{\leq} \mathrm{D}(f, g)$$

or

$$\mathrm{R}(f) + \mathrm{R}(g) \stackrel{?}{\leq} \mathrm{R}(f, g)$$

Note that as formulated above, we might get trivial lower bounds on $\mathrm{D}(f, g)$ (similarly $\mathrm{R}(f, g)$) simple because of the bit-size of the output. In an attempt to avoid such cases we at times consider instead a function of f and g, say, $\mathrm{D}(f) + \mathrm{D}(g) \stackrel{?}{\leq} \mathrm{D}(f \oplus g)$.

Example - Equality function: We know,

$$\mathrm{R}^{\mathrm{pub}}_{1/3}(EQ_n) = \Theta(1) \text{and} \, \mathrm{R}^{\mathrm{priv}}_{1/3}(EQ_n) = \Theta(\log n)$$

It is further obvious that $\mathrm{R}^{\mathrm{priv}}_{1/3}(EQ^{(l)}_n) \leq l\Theta(\log n)$
As a public coin protocol for $EQ^{(l)}_n$, we simply repeat the protocol for EQ_n taking care of errors on the way. It is trivial to observe that $\mathrm{R}^{\mathrm{pub}}_{1/3l}(EQ_n) = O(\log l)$.
Thus, $\mathrm{R}^{\mathrm{pub}}_{1/3}(EQ^{(l)}_n) = O(l \log l)$.
Using a result from problem set 2 then, we have here $\mathrm{R}^{\mathrm{priv}}_{1/3}(EQ^{(l)}_n) = O(l \log l + \log n)$
For $l = \log n$

$$\log n \log \log n \ll \log^2 n$$

Particular values of l then give us a non-trivial instance of the direct sum problem.

Example - LNE: Consider,

$$\mathrm{LNE}_{n,l}((x_1,\ldots,x_n),(y_1,\ldots,y_n)) = \begin{cases} 1 & \text{if } x_i \neq y_i \forall i \\ 0 & \text{otherwise} \end{cases}$$

Notice that $LNE_{n,l}$ is simply $\overline{EQ_n}^{(l)}$. With arguments entirely analogous to those in the first example it is easy to observe

$$\begin{aligned} \mathrm{R}^{\mathrm{priv}}_{1/3}(\overline{EQ_n}) &= \Theta(\log n) \\ \text{But, } \mathrm{R}^{\mathrm{priv}}_{1/3}(LNE_{n,l}) &= O(l+\log n) \\ &\ll l\,\mathrm{R}^{\mathrm{priv}}_{1/3}(\overline{EQ_n}) \\ &\quad (\text{for } l=\omega(1)) \end{aligned} \tag{14.2.1}$$

Notice however that in all the cases considered above we get only a log advantage on the trivial bounds. We will be considering later if we can get a better than logarithmic advantage here.

14.2.1 Deterministic and Non-deterministic case

With our main focus being the randomized case, we simply state known results for the deterministic and non-deterministic cases. Refer Section 4.1 of Kushilevitz and Nisan's book on Communication Complexity [KN97] for the relevant proofs.

Theorem 14.1. *For all boolean functions $f,g:\{0,1\}^n\times\{0,1\}^n\to\{0,1\}$*

- $\mathrm{N}(f\wedge g)\geq \mathrm{N}(f)+\mathrm{N}(g)-2\log n-O(1)$
- $\mathrm{N}(\wedge^l_{i=1}f)\geq l(\mathrm{N}(f)-\log n-O(1))$

Corollary 14.2. *For f,g as above*

- $\mathrm{D}(f\wedge g)\geq \sqrt{\mathrm{D}(f)}+\sqrt{\mathrm{D}(g)}-2\log n-O(1)$
- $\mathrm{D}(\wedge^l_{i=1}f)\geq l(\sqrt{\mathrm{D}(f)}-O(\log n)-O(1)$

14.2.2 Randomized Case

We have so far seen two ways to give lower bounds for randomized protocols.

- Discrepancy
- Information Complexity

Shaltiel in "Towards proving strong direct product theorems that Discrepancy preserves Direct sum" [Sha01] shows that the logarithm of discrepancy is additive over independent problem runs. Note however that we only have however a lower bound $\mathrm{D}^{\mu}_{\varepsilon} \geq \log \frac{1}{Disc}$ for the communication complexity involving discrepancy. This method, thus, does not yield itself to the direct sum problem due to the lack of any tight upper bound.
In the other case, we already have encountered the fact that "Information" respects direct sum. So far we have only seen that $\mathrm{D}^{\mu}_{\varepsilon} \geq IC_{\mu}(f)$. Some manner or reverse inequality would again be necessary before we can discuss Direct Sum in this framework. This is what motivates the methods of Message and Protocol Compression. Before that however we make a digression to reduce consideration from randomized complexity to distributional complexity.

14.2.3 Distributional Complexity

We had used Yao's lemma to argue $\mathrm{R}_{\varepsilon}(f) = max_{\mu}\, \mathrm{D}^{\mu}_{\varepsilon}(f)$. We have then,

$$(\mathrm{D}^{\mu^n}_{\varepsilon}(f^{(n)}) \geq n\, \mathrm{D}^{\mu}_{\varepsilon}(f)\ \forall \mu) \implies \mathrm{R}_{\varepsilon}(f^{(n)}) \geq n\, \mathrm{R}_{\varepsilon}(f)$$

by simply plugging in the witness of Yao's maximum in the left hand side. Direct sum in the distributional complexity case, as in the left hand side, would thus give us the corresponding result for randomized complexity too.

14.3 Message and Protocol Compression

We consider here a single step in the protocol

$$\underset{X}{A} \xrightarrow{M} \underset{Y}{B}$$

If there existed a $A \xrightarrow{z} B$ such that z and Y allow Bob to get M' distributed exactly like the M above then z encapsulates exactly the same amount of "information" as M. Message compression attempts to compress each message in the protocol to it's "information content".
Consider however the case where each step in the protocol has $o(1)$ information. In this case there is an $\omega(1)$ blow-up with respect to information even after message compression. In an attempt at getting around this flaw with message compression, we shift focus to "protocol compression".

Definition 14.3. *Let π be a randomized protocol using both public and private coins, and μ a distribution on the inputs. Define internal information content as,*

$$IC_{\mu}(\pi) = I[Y : \pi R|X] + I[X : \pi R|Y]$$

We call IC_{μ} "internal information content" to contrast it with "external information content", $I(XY|\pi R)$, that we have used so far in this course. IC_{μ} measures, intuitively, the amount of information each party gets about the other party's input. Notice that the random coins in the above expression are crucial. Without the random coins the transcript

could be completely independent of the inputs - consider the case when both parties use one bit of the public randomness to XOR their messages with. We make some remarks and observations about $IC_\mu(\pi)$ without proof.

Observation 14.4.

$$I(X : \pi R|Y) + I(Y : \pi R|X) \leq I[XY : \pi R]$$

Further, in case μ is a product distribution, we have equality above.

Observation 14.5.

$$IC_\mu(\pi) \leq |\pi|$$

Formalizing the notion of information content as above allows us to write down the **Protocol Compression problem** as considered in [BBCR10].
Given a randomized protocol π using both public and private coins, and a distribution μ on the inputs does there exist another protocol τ s.t.

- $|\tau| \approx IC_\mu(\pi)$
- at the termination of τ, Alice and Bob construct τ_A and τ_B which look like π.

14.3.1 Protocol Abstraction

It is possible to give an extension of the notion of a protocol tree from the deterministic case to the private coins randomized protocol. In complete analogy with the deterministic case, we have a binary tree with each internal owned by either Alice or Bob. In the randomized case however, the owner of a node possesses a probability distribution instead of a deterministic function of it's inputs. So for a node v owned by Alice (say), she has a probability distribution $P_{v,x}$ supported on the children of v, and entirely similarly for Bob. In execution of the protocol, at each step, the owner simply samples a node according to the distribution it holds.
A randomized protocol that uses both public and private coins can simply be realized as a distribution over private coins protocol trees.

14.3.2 Basic Idea in Protocol Compression

Alice and Bob first use public coins to sample a private coins protocol tree as mentioned above. They then each sample a path in the tree, guessing the distributions they themselves are not party to. Communication then transpires in an attempt to correct discrepancies in their guessed paths. While the protocol will be described formally in the next class, we give here the sampling technique employed for the initial guesses.

14.3.3 Path sampling

Use past notations to denote the probability distributions of the selected private coins protocol tree. We detail Alice's procedure here, Bob works entirely anaogously. For an

internal node v owned by Bob, we let $P_{v,Y}$ denote the distribution $\underset{y}{\text{avg}} P_{v,y}$. For each internal node v, use public coin randomness to pick $\kappa_v \in [0,1]$. For v owned by Alice, she chooses,

$$C^x(v) = \begin{cases} \text{left child of } v & \text{if } \kappa_v \leq P_{v,x} \\ \text{right child of } v & \text{otherwise} \end{cases}$$

If Bob owns v then,

$$C^x(v) = \begin{cases} \text{left child of } v & \text{if } \kappa_v \leq P_{v,Y} \\ \text{right child of } v & \text{otherwise} \end{cases}$$

Similarly Bob can choose $C^y(v)$ for each internal node v.
The key points in this protocol are:

- If Alice and Bob used the same distribution $P_{v,x,y}$ for some internal node, then they end up selecting the same child.
- In the case when, say, $P_{v,x} \neq P_{v,X}$ the probability of an inconsistency is related to $|P_{v,x} - P_{v,X}|$ i.e. the L^1-norm distance between the distributions.

In the coming lectures, we propose to show that (Expected number of disagreements) $\approx \log n$ towards a partial answer to the Protocol Compression Porblem.

References

[BBCR10] Boaz Barak, Mark Braverman, Xi Chen, and Anup Rao. *How to compress interactive communication.* In *Proc. 42nd ACM Symp. on Theory of Computing (STOC)*, pages 67–76. 2010. doi:10.1145/1806689.1806701.

[KN97] Eyal Kushilevitz and Noam Nisan. *Communication Complexity.* Cambridge University Press, 1997. doi:10.2277/052102983X.

[Sha01] Ronen Shaltiel. *Towards proving strong direct product theorems.* Technical Report TR01-009, Electronic Colloquium on Computational Complexity, 2001. eccc:TR01-009.