4. Applications to VLSI design and time-space tradeoffs

Lecturer: Jaikumar Radhakrishnan & Prahladh Harsha *Scribe: Sagnik Mukhopadhyay*

In today's lecture, we will see two applications of communication complexity. First, we will prove a lower bound on the physical limitations of VLSI chips use communication complexity. Then, we we will shift our attention to Turing machines, both multi-tape and one-tape variants, and we will prove time-space tradeoffs for multi-tape TMs and time lower bounds for one-tape TMs using communication complexity. This lecture is based on Chapters 8 and 12 of Kushilevtiz and Nisan's book on Communication Complexity [KN97].

4.1 $T\sqrt{A}$ lower bounds for VLSI

We begin with the $T\sqrt{A}$ lower bound for VLSI chips due to Thompson [Tho79], one of the earliest motivations for communication complexity. We abstract a chip as follows. We view a VLSI chip as a rectangular board of width a and length b (and thus with area $A = a \cdot b$) and on which some inputs are available. The goal of the chip is to compute a function $f : \{0,1\}^n \to \{0,1\}$. There is also an output port on which the output has to appear. Our model is the following. There can be many devices (gates) on the board which take inputs and give outputs and there are wires connecting them. We assume that the wires have unit thickness, i.e., only a wires can pass through a cut of the board of length a. The chip works in cycles, i.e., in each time step these devices put a bit on the wires and we will assume that within one time step the wire stabilizes and the device on the other end can reliably conclude what is the value of the bit in the wire and the communication proceeds. In the next timestep, each of the devices put a bit on the wires and so on. In no time step can a wire be driven from both sides. We assume that it takes T time steps to compute the function. In other words, after at most T time steps, the output appears at the output port. Equivalently, T is worst case number of cycles (over all inputs) for the computation to take place.

We lower bound the quantity $T\sqrt{A}$ using the best case communication complexity of f defined as follows.

Definition 4.1. *Let $f(x_1, ..., x_m)$ be a function. For any partition $(S, T = \overline{S})$ of the input locations $\{1, \ldots, m\}$, $\mathrm{D}^{S:T}(f)$ denotes the deterministic communication complexity of the function f when Alice is given the input bits corresponding to the set S (i.e, $(x_i, i \in S)$ and Bob is given the input bits corresponding to the set T, ie., $(x_i, i \in T)$. The* ***best case communication complexity*** *of f, denoted by $\mathrm{D}^{best}(f)$, is defined to be the minimum $\mathrm{D}^{S:T}(f)$ over all partitions $x_1, ..., x_m$ into two sets S, T of equal size.*

The assumption of S, T being of equal size is crucial, since otherwise one of Alice or Bob can have all the variables and the communication complexity then becomes 0. Furthermore, the sets S, T are decided before the input is provided, i.e., the partition depends on f, not on the inputs.

Theorem 4.2. *Suppose that a VLSI chip as described above of area A computes a function f in T time steps. Then*
$$T \cdot \sqrt{A} = \Omega(\mathrm{D}^{best}(f)).$$

Proof. The idea is to show that there exists a partition $(S, \overline{S})$ of the input bits based on the VLSI chip such that $\mathrm{D}^{S:\overline{S}}(f) = O(T\sqrt{A})$. We assume wlog. that the chip is of width a and length b where $a \leq b$. Hence, $a \leq \sqrt{A}$. We partition the board into two parts such that each part has $n/2$ devices. We claim that we can do it with a cut of length at most $a + 1$. This is because we can imagine sweeping the chip by a line parallel to the width along the length of the chip. We sweep till this line divides the chip into left and right parts until reaching the maximal location where the number of input devices to the left of the line is at most $n/2$. Then to make the number exactly $n/2$, we might need to add a *shoulder* to the cut. The partition $(S, \overline{S})$ of the n input bits between Alice and Bob is exactly the above partition (ie., Alice receives all input bits to the left of the partition and Bob all bits to the right of the above line).

Consider the following protocol to compute $f(x)$ when Alice is given $x|_S$ and Bob $x|_{\overline{S}}$. Alice and Bob simulate the VLSI chip exchanging in each time step the contents of the wires crossing the cut. At each time step, the maximum amount of communication that can happen is upper bounded by the size of cut, ie., $(a + 1)$. Hence, in T steps, the maximum amount of communication that happens is $T \cdot (a + 1) = O(T\sqrt{A})$. Hence, $\mathrm{D}^{best}(f) \leq \mathrm{D}^{S:\overline{S}}(f) = O(T\sqrt{A})$. □

We now ask the question whether there is any function which has large $\mathrm{D}^{best}(\cdot)$. Note that for the functions we have considered so far, equality (EQ), disjointness (DISJ), inner product (IP), (each of which require $\Omega(n)$ deterministic communication complexity), there exists a partition of the inputs such that the deterministic communication complexity corresponding to this partition (and hence D^{best}) is $O(1)$. What we will show is that a slight variant of the equality function EQ, namely *Shifted Equality* (denoted by SEQ), works. The function is defined as follows.

Definition 4.3. *For $x, y \in \{0,1\}^n$ and $0 \leq i \leq n-1$,* shifted equality *function,* $\mathsf{SEQ}_n(x, y, i) = 1$ *if for all* $0 \leq j < n, x_j = y_{(i+j) \pmod n}$

Informally, the function SEQ_n will output 1 on input (x, y, i) if the string x is equal to the string y shifted cyclically i bits to the right.

Claim 4.4. $\mathrm{D}^{best}(\mathsf{SEQ}_n) = \Omega(n)$

Proof. The total number of input bits is $2n + \log n$, the two strings being of length n each and the index i being of length $\log n$. Alice and Bob each gets an equal part of input. Wlog. we assume that Alice has at least $n/3$ fraction of the bits of x (call this set of bits A) and similarly Bob has at least $n/3$ fraction of the bits of y (call this B).

Now we consider a random rotation on y. Clearly, for a random i, since $|A|, |B| \geq n/3$, we have
$$\mathbb{E}\left[|A \cap (B + i)|\right] \geq \frac{n}{9}$$
where $B + i$ denotes the set B rotated i bits clockwise. Hence, there exists an i such that $|A \cap (B + i)| \geq n/9$. We fix such an i. For this i, let $A' \subset A$ and $B' \subset B$ be defined as

follows: $A' = A \cap (B+i)$ and $B' = (A-i) \cap B$. We now consider inputs in which all the bits except the ones in A' and B' are set to be zero. For these inputs, the problem reduces to an equality problem between A' and B', which are at least $n/9$ long. The lower bound now follows from the lower bound for $D(\mathsf{EQ}_{n/9})$.

$\square$

4.2 Turing machines

Our next application will be towards time-space tradeoffs for multi-tape Turing Machines (TMs) and time lower bounds for single tape TMs.

4.2.1 Multi-tape TMs: time-space tradeoffs

We assume the standard model of multi-tape TMs with one read-only input tape and k additional read-write tapes. The TM has a local view of these tapes, meaning, the machine can read only one symbol from each tape at which the head is located. Based on these, the machine decides what to write on the read-write tapes and which direction to move its heads. Each of the head can move only sequentially, i.e., no jumps are allowed.

A TM has time complexity $T(n)$ if the machine halts in $T(n)$ steps for all inputs of length n. The machine has space complexity $S(n)$ if the machine visits at most $S(n)$ cells of its k read-write tapes for all input of length n. We will assume wlog. that $T(n) \geq n$ and $S(n) \geq \log n$.

For example, consider the language of palindromes

$$\mathsf{PAL} = \{ww^R : w \in \{0,1\}^n\},$$

where w^R denotes the string $w_n w_{n-1} \ldots w_1$ if $w = w_1 w_2 \ldots w_n$. One way a TM can recognize this language is, given the input in the read-only tape, it first checks whether it is of even length. If it is, it copies the first half of the input to one of the read-write tape and compare the content of both tapes in opposite direction. The time taken in this process is linear, i.e., $\Theta(n)$. The space complexity is also linear, i.e., $\Theta(n)$.

An alternate approach is to move back and forth within the input string and match the symbols. In this case, we just need to store the index of the current position of the input head, making the space complexity $\Theta(\log n)$. But the time required is quadratic, i.e., $\Theta(n^2)$.

Can we have the best of the above two approaches? I.e., can we design a TM that decides PAL simultaneously in time $O(n)$ and space $O(\log n)$. We will use communication complexity to show that this is impossible.

Theorem 4.5. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. Let M be a multi-tape TM which accepts the set*

$$\{x0^n y : |x| = |y| = n, f(x,y) = 1\}$$

and rejects the set

$$\{x0^n y : |x| = |y| = n, f(x,y) = 0\}$$

and runs in time $T(n)$ and space $S(n)$. Then

$$T(n) \cdot S(n) = \Omega(n \cdot \mathrm{D}(f)).$$

Proof. We will prove the upper bound by designing a protocol based on the machine M. Alice and Bob, on input x and y respectively, try to simulate M on the input $x0^n y$. Note that Alice does not have y and Bob does not have x. Alice begins the simulation and continues as long as the input head is in the region of $x0^n$. When the head makes a transition into the region of y, Alice transfers control of M to Bob by sending him the contents of all the work tape (at most $O(S(n))$ bits long), and Bob resumes the simulation. We call this as a shift. Similarly Bob continues simulation and makes a shift when the head makes a transition into the region of x (i.e, he performs the simulation as long as the input head remains in the region $0^n y$. It is to be noted that in both cases, for making the shift, the head has to move through n 0's. And they continue in this process till the machine M terminates.

Let us now bound the communication cost of the above protocol. Clearly, the amount of communication at each shift is bounded by $S(n)$. To count the number of shifts, we will exploit the fact that the input head moves sequentially, and hence it has to cover n 0's between shifts. So the total number of shifts is bounded by $T(n)/n$. Hence we conclude that the total communication is $O(S(n) \cdot T(n)/n)$. $\square$

We now use the above theorem to show that the product of time and space for any TM deciding PAL is at least quadratic.

Corollary 4.6. $T_{\mathsf{PAL}}(n) \cdot S_{\mathsf{PAL}}(n) = \Omega(n^2)$.

Proof Sketch. Consider a sub-language of PAL, consisting of strings ww^R for which the last $n/3$ fraction of bits in w are all 0's. Each such string is of the form $x0^{n'}x^R$ where $n' = 2n/3$ and $x \in \{0,1\}^{n'}$. Checking that $x0^{n'}y^R \in \mathsf{PAL}$ is equivalent to checking iff $\mathsf{EQ}_{2n/3}(x,y) = 1$. The corollary follows since $\mathrm{D}(\mathsf{EQ}_{2n/3}) = \Omega(n)$. $\square$

4.2.2 One-tape Turing machine: time lower bounds

We will now use a similar approach to show time lower bounds for one-tape TMs. Recall that a one-tape TM has only one read-write tape on which the input is written. It has no additional work tapes. Before describing the lower bound, we need the following notions from communication complexity.

In randomized communication complexity, there are two types of protocol based on the error.

1. *Bounded error protocol:* Here we consider protocols where the answer might be wrong with a non-zero but small (say 1/4) probability, but the length of the protocol is a fixed deterministic quantity (dependent on the input). The bounded error randomized communication complexity of f, denoted by $\mathrm{R}^{\mathrm{pub}}_{1/4}(f)$, is the worst-case (over all pairs of inputs) number of bits exchanged by the best bounded-error protocol that computes the function f.

2. *Zero error protocol:* Here, the protocol always outputs the correct answer, but the length of the protocol is a random variable. The zero-error randomized communication complexity of f, denoted by $\mathrm{R}^{\mathrm{pub}}_0(f)$, is the worst-case (over all pairs of inputs) expected number of bits exchanged by the best zero-error protocol that computes the function f.

We could consider private coins versions of the above protocols too.

For example, $\mathrm{R}^{\mathrm{pub}}_{1/4}(\mathsf{EQ}_n) = O(1)$ whereas $\mathrm{R}^{\mathrm{pub}}_{0}(\mathsf{EQ}_n) = \Omega(n)$. This follows from the observation that $\mathrm{R}^{\mathrm{pub}}_{0}(f)$ is bounded below by the logarithm of the number of rectangles in a monochromatic cover, i.e., $max(N(f), N(\overline{f}))$ (see Problem Set 1).

We now state the theorem which gives a time lower bound for 1-tape TMs using communication complexity.

Theorem 4.7. *Let $f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$ be a function. Let M be a one-tape TM which accepts the set*

$$\{x0^n y : |x| = |y| = n, f(x,y) = 1\}$$

and rejects the set

$$\{x0^n y : |x| = |y| = n, f(x,y) = 0\}$$

and runs in time $T(n)$. Then

$$T(n) = \Omega(n \cdot \mathrm{R}^{pub}_0(f)).$$

Proof. Consider the following zero-error public coins protocol for f. On input x to Alice and y to Bob, they simulate the machine M on input $x0^n y$. First, using public randomness, they determine a dividing point within the 0 region of the input. Alice performs the simulation as long as the head is to the left of this point and Bob when the head is to the right of this point. Each time the head transitions across this point, one party transfers control to the other by sending the state, which is $O(1)$ bits. Here we exploit the fact that the TM is one-tape, hence the only thing to be transferred is the state of the head (specifically, there is no work-tape content).

For $i \in [n]$, let c_i be the number of times the head crosses the i-location in the region 0^n. Clearly, $\sum_{i=1}^n c_i \le T(n)$. On the other hand, the expected number of transitions between Alice and Bob is exactly $\mathbb{E}_i[c_i]$ where i is chosen randomly. Thus, the expected number of transitions is at most $T(n)/n$. Hence the expected number of bits communicated is $O(1) \cdot T(n)/n = O\left(\frac{T(n)}{n}\right)$. □

The above theorem combined with the fact that $\mathrm{R}^{\mathrm{pub}}_{0}(\mathsf{EQ}_n) = \Omega(n)$ gives us the following.

Corollary 4.8. *Any one-tape TM that solves* PAL *requires $\Omega(n^2)$ time.*

4.2.3 Open questions

Since communication complexity is at most n for any problem, the approach indicated in Theorem 4.5 and Theorem 4.7 can give at most a quadratic lower bound. On the other hand, we do not know of any technique that yields a better than quadratic lower bound for either problem. With respect to Theorem 4.7, it is open if the simulation can be made deterministic. More importantly, we do not know of any lower bound (better than linear) for computing an explicit function on two-tape TMs or any general model of computation other than 1-tape TMs.

References

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

[Tho79] Clark D. Thompson. *Area-time complexity for VLSI*. In *Proc. 11th ACM Symp. on Theory of Computing (STOC)*, pages 81–88. 1979. doi:10.1145/800135.804401.