Special Topics in Complexity Theory, Fall 2017. Instructor: Emanuele Viola

1 Lectures 12-13, Scribe: Giorgos Zirdelis

In these lectures we study the communication complexity of some problems on groups. We give the definition of a protocol when two parties are involved and generalize later to more parties.

Definition 1. A 2-party c-bit *deterministic* communication protocol is a depth-c binary tree such that:

- the leaves are the output of the protocol
- each internal node is labeled with a party and a function from that party's input space to $\{0,1\}$

Computation is done by following a path on edges, corresponding to outputs of functions at the nodes.

A public-coin *randomized* protocol is a distribution on deterministic protocols.

2 2-party communication protocols

We start with a simple protocol for the following problem.

Let G be a group. Alice gets $x \in G$ and Bob gets $y \in G$ and their goal is to check if $x \cdot y = 1_G$, or equivalently if $x = y^{-1}$.

There is a simple deterministic protocol in which Alice simply sends her input to Bob who checks if $x \cdot y = 1_G$. This requires $O(\log |G|)$ communication complexity.

We give a randomized protocol that does better in terms on communication complexity. Alice picks a random hash function $h : G \to \{0,1\}^\ell$. We can think that both Alice and Bob share some common randomness and thus they can agree on a common hash function to use in the protocol. Next, Alice sends $h(x)$ to Bob, who then checks if $h(x) = h(y^{-1})$.

For $\ell = O(1)$ we get constant error and constant communication.

3 3-party communication protocols

There are two ways to extend 2-party communication protocols to more parties. We first focus on the Number-in-hand (NIH), where Alice gets x, Bob gets y, Charlie gets z, and they want to check if $x \cdot y \cdot z = 1_G$. In the NIH setting the communication depends on the group G.

3.1 A randomized protocol for the hypercube

Let $G = (\{0,1\}^n, +)$ with addition modulo 2. We want to test if $x+y+z = 0^n$. First, we pick a linear hash function h, i.e. satisfying $h(x+y) = h(x) + h(y)$. For a uniformly random $a \in \{0,1\}^n$ set $h_a(x) = \sum a_i x_i \pmod 2$. Then,

- Alice sends $h_a(x)$
- Bob send $h_a(y)$
- Charlie accepts if and only if $\underbrace{h_a(x) + h_a(y)}_{h_a(x+y)} = h_a(z)$

The hash function outputs 1 bit. The error probability is 1/2 and the communication is $O(1)$. For a better error, we can repeat.

3.2 A randomized protocol for $\mathbb{Z}_m$

Let $G = (\mathbb{Z}_m, +)$ where $m = 2^n$. Again, we want to test if $x + y + z = 0 \pmod m$. For this group, there is no 100% linear hash function but there are almost linear hash function families $h : \mathbb{Z}_m \to \mathbb{Z}_\ell$ that satisfy the following properties:

1. $\forall a, x, y$ we have $h_a(x) + h_a(y) = h_a(x+y) \pm 1$
2. $\forall x \neq 0$ we have $\Pr_a[h_a(x) \in \{\pm 2, \pm 1, 0\}] \leq 2^{-\Omega(\ell)}$
3. $h_a(0) = 0$

Assuming some random hash function h (from a family) that satisfies the above properties the protocol works similar to the previous one.

- Alice sends $h_a(x)$

- Bob sends $h_a(y)$
- Charlie accepts if and only if $h_a(x) + h_a(y) + h_a(z) \in \{\pm 2, \pm 1, 0\}$

We can set $\ell = O(1)$ to achieve constant communication and constant error.

Analysis

To prove correctness of the protocol, first note that $h_a(x)+h_a(y)+h_a(z) = h_a(x+y+z) \pm 2$, then consider the following two cases:

- if $x + y + z = 0$ then $h_a(x + y + z) \pm 2 = h_a(0) \pm 2 = 0 \pm 2$
- if $x + y + z \neq 0$ then $\Pr_a[h_a(x + y + z) \in \{\pm 2, \pm 1, 0\}] \leq 2^{-\Omega(\ell)}$

It now remains to show that such hash function families exist.

Let a be a random odd number modulo 2^n. Define

$$h_a(x) := (a \cdot x \gg n - \ell) \pmod{2^\ell}$$

where the product $a \cdot x$ is integer multiplication. In other words we output the bits $n - \ell + 1, n - \ell + 2, \ldots, n$ of the integer product $a \cdot x$.

We now verify that the above hash function family satisfies the three properties we required above.

Property (3) is trivially satisfied.

For property (1) we have the following. Let $s = a \cdot x$ and $t = a \cdot y$ and $u = n - \ell$. The bottom line is how $(s \gg u) + (t \gg u)$ compares with $(s + t) \gg u$. In more detail we have that,

- $h_a(x + y) = ((s + t) \gg u) \pmod{2^\ell}$
- $h_a(x) = (s \gg u) \pmod{2^\ell}$
- $h_a(x) = (t \gg u) \pmod{2^\ell}$

Notice, that if in the addition $s + t$ the carry into the $u + 1$ bit is 0, then

$$(s \gg u) + (t \gg u) = (s + t) \gg u$$

otherwise

$$(s \gg u) + (t \gg u) + 1 = (s + t) \gg u$$

which concludes the proof for property (1).

Finally, we prove property (2). We start by writing $x = s \cdot 2^c$ where s is odd. Bitwise, this looks like $(\cdots\cdots 1\,\underbrace{0\cdots 0}_{c \text{ bits}})$.

The product $a{\cdot}x$ for a uniformly random a, bitwise looks like $(\textit{uniform}\,1\,\underbrace{0\cdots 0}_{c \text{ bits}})$.

We consider the two following cases for the product $a \cdot x$:

1. If $a{\cdot}x = (\underbrace{\textit{uniform}\ 1\ \overbrace{00}^{2\ \textit{bits}}}_{\ell\ \textit{bits}}\cdots 0)$, or equivalently $c \geq n-\ell+2$, the output never lands in the bad set $\{\pm 2, \pm 1, 0\}$ (some thought should be given to the representation of negative numbers – we ignore that for simplicity).

2. Otherwise, the hash function output has $\ell - O(1)$ uniform bits. Again for simplicity, let $B = \{0, 1, 2\}$. Thus,

   $$\Pr[\text{output} \in B] \leq |B| \cdot 2^{-\ell+O(1)}$$

   In other words, the probability of landing in any small set is small.

4 Other groups

What happens in other groups? Do we have an almost linear hash function for 2×2 matrices? The answer is negative. For $SL_2(q)$ and A_n the problem of testing equality with 1_G is hard.

We would like to rule out randomized protocols, but it is hard to reason about them directly. Instead, we are going to rule out deterministic protocols on random inputs. For concreteness our main focus will be $SL_2(q)$.

First, for any group element $g \in G$ we define the distribution on triples, $D_g := (x, y, (x \cdot y)^{-1}g)$, where $x, y \in G$ are uniformly random elements. Note the product of the elements in D_g is always g.

Towards a contradiction, suppose we have a randomized protocol P for the $xyz =^? 1_G$ problem. In particular, we have

$$\Pr[P(D_1) = 1] \geq \Pr[P(D_h) = 1] + \frac{1}{10}.$$

This implies a deterministic protocol with the same gap, by fixing the randomness.

We reach a contradiction by showing that for every *deterministic* protocols P using little communication (will quantify later), we have

$$|\Pr[P(D_1) = 1] - \Pr[P(D_h) = 1]| \leq \frac{1}{100}.$$

We start with the following lemma, which describes a protocol using product sets.

Lemma 1. (The set of accepted inputs of) A deterministic c-bit protocol can be written as a disjoint union of 2^c "rectangles," that is sets of the form $A \times B \times C$.

Proof. (sketch) For every communication transcript t, let $S_t \subseteq G^3$ be the set of inputs giving transcript t. The sets S_t are disjoint since an input gives only one transcript, and their number is 2^c, i.e. one for each communication transcript of the protocol. The rectangle property can be proven by induction on the protocol tree. □

Next, we show that these product sets cannot distinguish these two distributions D_1, D_h, and for that we will use the pseudorandom properties of the group G.

Lemma 2. For all $A, B, C \subseteq G$ and we have

$$|\Pr[A \times B \times C(D_1) = 1] - \Pr[A \times B \times C(D_h) = 1]| \leq \frac{1}{d^{\Omega(1)}}.$$

Recall the parameter d from the previous lectures and that when the group G is $SL_2(q)$ then $d = |G|^{\Omega(1)}$.

Proof. Pick any $h \in G$ and let x, y, z be the inputs of Alice, Bob, and Charlie respectively. Then

$$\Pr[A \times B \times C(D_h) = 1] = \Pr[(x, y) \in A \times B] \cdot \Pr[(x \cdot y)^{-1} \cdot h \in C | (x, y) \in A \times B]$$

If either A or B is small, that is $\Pr[x \in A] \leq \epsilon$ or $\Pr[y \in B] \leq \epsilon$, then also $\Pr[P(D_h) = 1] \leq \epsilon$ because the term $\Pr[(x, y) \in A \times B]$ will be small. We will choose ϵ later.

Otherwise, A and B are large, which implies that x and y are uniform over at least $\epsilon|G|$ elements. Recall from Lecture 9 that this implies $\|x \cdot y - U\|_2 \leq \|x\|_2 \cdot \|y\|_2 \cdot \sqrt{\frac{|G|}{d}}$, where U is the uniform distribution.

By Cauchy–Schwarz we obtain,

$$\|x \cdot y - U\|_1 \leq |G| \cdot \|x\|_2 \cdot \|y\|_2 \cdot \sqrt{\frac{1}{d}} \leq \frac{1}{\epsilon} \cdot \frac{1}{\sqrt{d}}.$$

The last inequality follows from the fact that $\|x\|_2, \|y\|_2 \leq \sqrt{\frac{1}{\epsilon|G|}}$.

This implies that $\|(x \cdot y)^{-1} - U\|_1 \leq \frac{1}{\epsilon} \cdot \frac{1}{\sqrt{d}}$ and $\|(x \cdot y)^{-1} \cdot h - U\|_1 \leq \frac{1}{\epsilon} \cdot \frac{1}{\sqrt{d}}$, because taking inverses and multiplying by h does not change anything. These two last inequalities imply that,

$$\Pr[(x \cdot y)^{-1} \in C | (x, y) \in A \times B] = \Pr[(x \cdot y)^{-1} \cdot h \in C | (x, y) \in A \times B] \pm \frac{2}{\epsilon} \frac{1}{\sqrt{d}}$$

and thus we get that,

$$\Pr[P(D_1) = 1] = \Pr[P(D_h) = 1] \pm \frac{2}{\epsilon} \frac{1}{\sqrt{d}}.$$

To conclude, based on all the above we have that for all ϵ and independent of the choice of h, it is either the case that

$$|\Pr[P(D_1) = 1] - \Pr[P(D_h) = 1]| \leq 2\epsilon$$

or

$$|\Pr[P(D_1) = 1] - \Pr[P(D_h) = 1]| \leq \frac{2}{\epsilon} \frac{1}{\sqrt{d}}$$

and we will now choose the ϵ to balance these two cases and finish the proof:

$$\frac{2}{\epsilon} \frac{1}{\sqrt{d}} = 2\epsilon \Leftrightarrow \frac{1}{\sqrt{d}} = \epsilon^2 \Leftrightarrow \epsilon = \frac{1}{d^{1/4}}.$$

□

The above proves that the distribution D_h behaves like the uniform distribution for product sets, for all $h \in G$.

Returning to arbitrary deterministic protocols P, write P as a union of 2^c disjoint rectangles by the first lemma. Applying the second lemma and summing over all rectangles we get that the distinguishing advantage of P is at most $2^c/d^{1/4}$. For $c \leq (1/100)\log d$ the advantage is at most $1/100$ and thus we get a contradiction on the existence of such a correct protocol. We have concluded the proof of this theorem.

Theorem 3. Let G be a group, and d be the minimum dimension of an irreducible representation of G. Consider the 3-party, number-in-hand communication protocol $f : G^3 \to \{0,1\}$ where $f(x,y,z) = 1 \Leftrightarrow x \cdot y \cdot z = 1_G$. Its randomized communication complexity is $\Omega(\log d)$.

For $SL_2(q)$ the communication is $\Omega(\log |G|)$. This is tight up to constants, because Alice can send her entire group element.

For the group A_n the known bounds on d yield communication $\Omega(\log\log |G|)$. This bound is tight for the problem of distinguishing D_1 from D_h for $h \neq 1$, as we show next. The identity element 1_G for the group A_n is the identity permutation. If $h \neq 1_G$ then h is a permutation that maps some element $a \in G$ to $h(a) = b \neq a$. The idea is that the parties just need to "follow" a, which is logarithmically smaller than G. Specifically, let x, y, z be the permutations that Alice, Bob and Charlie get. Alice sends $x(a) \in [n]$. Bob gets $x(a)$ and sends $y(x(a)) \in [n]$ to Charlie who checks if $z(y(x(a))) = 1$. The communication is $O(\log n)$. Because the size of the group is $|G| = \Theta(n!) = \Theta\left(\left(\frac{n}{e}\right)^n\right)$, the communication is $O(\log\log |G|)$.

This is also a proof that d cannot be too large for A_n, i.e. is at most $(\log |G|)^{O(1)}$.

5 More on 2-party protocols

We move to another setting where a clean answer can be given. Here we only have two parties. Alice gets $x_1, x_2, \ldots, x_n$, Bob gets $y_1, y_2, \ldots, y_n$, and they want to know if $x_1 \cdot y_1 \cdot x_2 \cdot y_2 \cdots x_n \cdot y_n = 1_G$.

When G is abelian, the elements can be reordered as to check whether $(x_1 \cdot x_2 \cdots x_n) \cdot (y_1 \cdot y_2 \cdots y_n) = 1_G$. This requires constant communication (using randomness) as we saw in Lecture 12, since it is equivalent to the check $x \cdot y = 1_G$ where $x = x_1 \cdot x_2 \cdots x_n$ and $y = y_1 \cdot y_2 \cdots y_n$.

We will prove the next theorem for non-abelian groups.

Theorem 1. For every non-abelian group G the communication of deciding if $x_1 \cdot y_1 \cdot x_2 \cdot y_2 \cdots x_n \cdot y_n = 1_G$ is $\Omega(n)$.

Proof. We reduce from unique disjointness, defined below. For the reduction we will need to encode the And of two bits $x, y \in \{0,1\}$ as a group product. (This question is similar to a puzzle that asks how to hang a picture on the wall with two nails, such that if either one of the nails is removed, the picture will fall. This is like computing the And function on two bits, where both bits (nails) have to be 1 in order for the function to be 1.) Since G is non-abelian, there exist $a, b \in G$ such that $a \cdot b \neq b \cdot a$, and in particular $a \cdot b \cdot a^{-1} \cdot b^{-1} = h$ with $h \neq 1$. We can use this fact to encode And as

$$a^x \cdot b^y \cdot a^{-x} \cdot b^{-y} = \begin{cases} 1, & \text{if And(x,y)=0} \\ h, & \text{otherwise} \end{cases} .$$

In the disjointness problem Alice and Bob get inputs $x, y \in \{0,1\}^n$ respectively, and they wish to check if there exists an $i \in [n]$ such that $x_i \wedge y_i = 1$. If you think of them as characteristic vectors of sets, this problem is asking if the sets have a common element or not. The communication of this problem is $\Omega(n)$. Moreover, in the variant of this problem where the number of such i's is 0 or 1 (i.e. unique), the same lower bound $\Omega(n)$ still applies. This is like giving Alice and Bob two sets that either are disjoint or intersect in exactly one element, and they need to distinguish these two cases.

Next, we will reduce the above variant of the set disjointness to group products. For $x, y \in \{0,1\}^n$ we product inputs for the group problem as follows:

$$\begin{aligned} x &\longrightarrow (a^{x_1}, a^{-x_1}, \ldots, a^{x_n}, a^{-x_n}) \\ y &\longrightarrow (b^{y_1}, b^{-y_1}, \ldots, b^{y_n}, b^{-y_n}). \end{aligned}$$

Now, the product $x_1 \cdot y_1 \cdot x_2 \cdot y_2 \cdots x_n \cdot y_n$ we originally wanted to compute becomes

$$\underbrace{a^{x_1} \cdot b^{y_1} \cdot a^{-x_1} \cdot b^{-y_1}}_{\text{1 bit}} \cdot \cdots \cdots a^{x_n} \cdot b^{y_n} \cdot a^{-x_n} \cdot b^{-y_n}.$$

If there isn't an $i \in [n]$ such that $x_i \wedge y_i = 1$, then each product term $a^{x_i} \cdot b^{y_i} \cdot a^{-x_i} \cdot b^{-y_i}$ is 1 for all i, and thus the whole product is 1.

Otherwise, there exists a unique i such that $x_i \wedge y_i = 1$ and thus the product will be $1 \cdots 1 \cdot h \cdot 1 \cdots 1 = h$, with h being in the i-th position. If Alice and Bob can test if the above product is equal to 1, they can also solve the unique set disjointness problem, and thus the lower bound applies for the former. □

We required the uniqueness property, because otherwise we might get a product h^c that could be equal to 1 in some groups.