CS 2429 - Foundations of Communication Complexity

Lecturer: Toniann Pitassi

1 Applications of Communication Complexity

There are many applications of communication complexity. In our survey article, "The Story of Set Disjointness" we give many applications via reductions to set disjointness (both 2-party as well as NOF model). Applications discussed in our survey article:

(1.) Streaming

(2.) Data Structures

(3.) Circuit Complexity

(4.) Proof Complexity

(5.) Game Theory

(6.) Quantum Computation

Below we discuss another application that is different than above. The essential difference is that it relies on the communication complexity of computing a certain *relation*, rather than a function.

1.1 Circuit Depth via Communication Complexity

In order to get circuit lower bounds, we need to extend our notion of 2-party communication complexity so that it can compute relations.

Definition A relation R is a subset $R \subseteq X \times Y \times Z$

Given a relation R the cc problem associated with R follows:
Alice gets $x \in X$
Bob gets $y \in Y$
Alice and Bob must both compute (and output) some z s.t. $(x, y, z) \in R$

A protocol for relations is the same as a protocol for functions, in each step it must specifiy which party sends a message and the value of that message.

Note that for a given relation there may be more than on z satisfying the above property, Alice and Bob only need to give one such z. In general, lower bounds are harder to prove for relations as we need to show it is hard for Alice and Bob to compute *any* z.

Definition For any boolean function $f : \{0,1\}^n \to \{0,1\}$ and $X = f^{-1}(1)$, $Y = f^{-1}(0)$. We define $R_f \subseteq X \times Y \times \{1, 2, ...n\}$ to be the associated relation where,

- $R_f = \{(x, y, i) | x \in X, y \in Y, x_i \neq y_i\}$

R_f is the set of all (x, y, i) where $f(x) = 1$, $f(y) = 0$ and x and y differ on bit i. Similarly if f is monotone then

- $M_f \subseteq X \times Y \times \{1, 2, ...n\}$ is the set of all (x, y, i) such that $x \in X$, $y \in Y$ and $x_i = 1$, $y_i = 0$.

(Recall that for a monotone boolean function f, $f(x) = 1$ implies that for all x' where $x'_i \geq x_i$ on every i, x' is also a 1 of the function.)

Communication complexity lower bounds on M_f give bounds on monotone circuit depth of f and lower bounds on R_f give circuit depth bounds for general circuits.

Let $d(f)$ and $d^{monotone}(f)$ denote the min depth of a circuit computing f over $\wedge$, $\vee$, $\neg$, and the min depth of a monotone circuit computing f over $\wedge$, $\vee$ respectively. In both cases the circuits must have bounded fan-in.

Theorem 1 *(Karchmer and Widerson '80s)*

1. *For every boolean function $f : \{0,1\}^n \to \{0,1\}$, $cc(R_f) = d(f)$*
2. *For f monotone, $cc(M_f) = d^{monotone}(f)$.*

For formulas it is known that $2^{d(f)} =$ formula-size(f) so proving lower bounds on communication complexity of relations is also equivalent to proving formula size lower bounds.

It is a major open problem to get even super log-depth lower bounds for the general case. But for the monotone case the method above has been used to show that $NC^i_{monotone} \neq NC^{i+1}_{monotone}$ for all i [see Theorem 2 and 3].

Proof of Theorem 1 "⇒"

Let C be a circuit for f, depth$(C) = d$. We can assume that all the negations in the circuit are at the leaves. (If not, the negations can be pushed to the leaves without affecting depth in any circuit by repeated application of DeMorgan's laws.)

We want to use the circuit to obtain a protocol for R_f.

The protocol will involve Alice and Bob taking a particular path down the circuit with Alice, deciding the branch to take at OR gates and Bob deciding at AND gates. As long as the two parties maintain the invariant that at each subnode v $C_v(x) = 1$ while $C_v(y) = 0$ then the leaf reached is a bit i where $x_i \neq y_i$.

The protocol follows:

Starting from the top of the circuit, for each each node v with children v_L, v_R

if the gate is an OR Alice says 0 if $C_{v_L}(x) = 1$ and 1 otherwise.

if the gate is an AND Bob says 0 $C_{v_L}(y) = 1$ and 1 otherwise.

At the end of the exchange, both Alice and bob recurse on v_L if the message sent was 0 and v_R if the message sent was 1.

Clearly at the top of the circuit, for any inputs (x, y), $C(x) \neq C(y)$. Suppose at some point during the protocol Alice and Bob are at some inner node v where $C_v(x) \neq C_v(y)$.

Case 1 v is an or node.

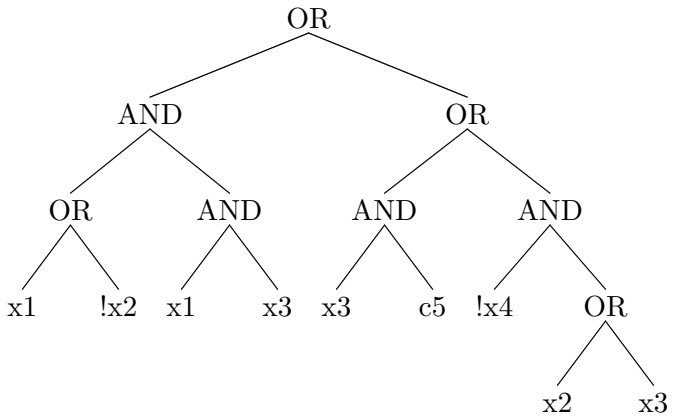
Then $C_v(y) = 0$ implies that both $C_{v_L}(y)$ and $C_{v_R}(y)$ are also 0. By choosing the subcircuit for which her input evaluates to 1, Alice ensures that the recursion continues on a subcircuit where the two inputs differ.

Case 2 v is an and node.

Likewise, $C_v(x) = 1 \Rightarrow C_{v_L}(x) = C_{v_R}(x) = 1$ so by choosing the subcircuit for which his input evaluates to 0 Bob can also maintain the above invariant.

By induction, when the protocol reaches a leaf, both A and B know an i at which their inputs differ. The total number of bits sent is bounded by the depth of the circuit. If C was monotone the same protocol reaches a left where $x_i = 1$.

Example



Suppose Alice and Bob have inputs (01101) and (01010) respectively. Then on the circuit above the sequence of bits sent would be.

Alice : 0 (go right)

Bob : 1(go left)

Alice : 0 (go left)

At which point they reach x_3 a bit on which they differ.

Proof of Theorem 1 "⇐"

Given a protocol for R_f we can construct a circuit computing f of bounded depth.

Consider a protocol tree T for R_f. Convert T into a circuit as follows:

1. For each node where the message is sent by Alice, replace the node with an OR gate
2. For each node where the message is sent by Bob, replace the node with an AND gate
3. At each leaf of the protocol tree, with associated monochromatic rectangle $A \times B$ and input bit i

   Claim Exactly one of the following hold

(a) $\forall \alpha \in A$, $\alpha_i = 1$ and $\forall \beta \in B$, $\beta_i = 0$

(b) $\forall \alpha \in A$, $\alpha_i = 0$ and $\forall \beta \in B$, $\beta_i = 1$

Assign the leaves in case (a) to be z_i and and the leaves in case (b) to be $\bar{z_i}$.

Given the claim we can prove by induction that the circuit thus constructed calculates $f(z)$.

Proof of Claim

Let $\alpha \in A$, $\alpha_i = \sigma$. Then for every $\beta \in B$, $\beta_i = \bar{\sigma}$ which in turn implies that $\forall \alpha \in A$, $\alpha_i = \sigma$.

2 Monotone Circuit Lower Bounds

For the rest of this lecture, we will how how to prove monotone depth circuit lower bounds. We will loosely follow the paper "Communication Lower Bounds via Critical Block Sensitivity" by Goos and Pitassi.

We first give a brief history of monotone circuit depth lower bounds. The first lower bounds were due to Razborov, who proved that any circuit for clique requires exponential-size circuits. In particular this implies a n^ϵ depth lower bound, showing that monotone P is not equal to monotone NP. Later, Karchmer-Wigderson introduced the connection (equivalence) mentioned above between circuit depth and communication complexity, and using this technique they proved that the st-connectivity function requires monotone depth $O(\log n)^2$, thus separating monotone NC^1 from monotone NC^2. Raz and McKenzie then showed that for every i, monotone NC^i is not equal to monotone NC^{i+1}.

Both Karchmer-Wigderson as well as Raz-McKenzie give rather complicated arguments. In another paper by Raz and Wigderson, it is shown that the depth of any monotone circuit for matching is n^ϵ via a reduction to set disjointness. Today and next lecture we will present a technique that will give the best monotone depth lower bounds known (depth $n/\log n$) as well as separating NC^i from NC^{i+1}. We use many ideas from Raz-McKenzie as well as new ideas.

The high level overview of our proof is as follows. First, we will switch gears and define a different kind of search problem, that we will call a CNF search problem, as well as a "lifted" CNF search problem. The lifted CNF search problem is a communication complexity problem. Secondly, we will show that from a lifted CNF search problem, one can define a related monotone function, such that lower bounds on the communication complexity of the lifted CNF search problem imply lower bounds on the related monotone function.

Then we will show how to prove lower bounds for lifted CNF search problems via a reduction to set disjointness.

2.1 CNF search problems and lifted CNF search problems

Let F be an unsatisfiable CNF formula over variables $z_1, \ldots, z_m$. The search problem associated with F, $S(F)$ is the following problem: the input is an assignment γ to $z_1, \ldots, z_m$, and the output should be a clause $C_i \in F$ that is falsified by γ.

A hard CNF search problem, intuitively, will be one where we have to look at many variables in order to pinpoint a clause that is false. For now, we can think of decision tree complexity as a good intuitive measure of hardness. (Although our actual measure of hardness will be a

more complicated measure called criticial block sensitivity.) As a simple example, suppose that $F = (x_1 \vee x_2)(\neg x_1 \vee x_2)(\neg x_2)F'$, where F' is a conjunction of clauses. In this case, F is "easy" because by just looking at variables x_1 and x_2, we can always find a violated clause. On the other hand, for other unsatisfiable formulas, this will not be the case, and in the worst case we may have to look at linearly many of the variables in order to find a clause that is made false.

Looking ahead, we will want to find a k-CNF unsatisfiable formula, or a kCSP unsatisfiable formula that is hard. One example is the Tseitin formulas. Let G be an undirected k-regular graph on n vertices, where n is odd. Then F_G is a kCSP whose variables are the underlying edges of G. The constraints of F_G are as follows. For every vertex v in G, we have a constraint that specifies that the mod 2 sum of the edges incident with v is odd. Since G has an odd number of vertices, there are an odd number of constraints, and thus the formula F_G is unsatisiable. (Each constraint corresponds to a parity equation; adding all parity equations together mod 2, we get $0 = 1$ since every edge occurs exactly twice on the left-side, and since there are an odd number of constraints the right-side will be 1.) If G is highly expanding (highly connected), then F_G will be a hard unsatisfiable kCSP in the sense that the decision tree complexity of the search problem associated with F_G will be large (and so will the critical block sensitivity) as we will see later.

Given an unsatisfiable kCNF or kCSP F over $z_1, \ldots, z_n$, we now define a lifted CNF search problem as follows. Let $g : \{0,1\}^c \times \{0,1\}^c \to \{0,1\}$ be a function (to be defined later). We refer to g as a "gadget" and you should think of c as very small. When $c = 2$, g is a boolean function of 4 bits. Alice's input will be $x_1, \ldots, x_n$, $n = cm$, where we will think of x as being divided up into m blocks, each of size c. Similarly Bob's input willbe $y_1, \ldots, y_n$, again where we think of y as being divided up into m blocks, each of size c. The lifted search problem is: given an assignment α, β to x and y respectiviely, solve $S(F)$ on input $z_1 = g(x^1, y^1), z_2 = g(x^2, y^2), ..., z_m = g(x^m, y^m)$, where x^i, y^i denotes the i^{th} block of variables in x, y respectively. We denote this problem by $S(F) o g^n$.

The idea is that if we start with a CNF search problem F of high decision tree complexity, in the lifted search problem as long as g is a good gadget, we will be forcing the players to communicate some bits in order to learn any one z_i value, and thus, the communication complexity of the lifted problem should roughly correspond to the decision tree complexity of the (unlifted) CNF search problem.

We will now show that lower bounds for a lifted CNF search problem imply depth lower bounds for a corresponding monotone function, as desired.

Theorem 2 *Let $g : X \times Y \to \{0,1\}$ be a two-party gadget, $|X| = 2^x, |Y| = 2^c$, and let F be an unsatisfiable kCSP on m variables and m' constraints. There is an explicit construction of a monotone function $f : \{0,1\}^n \to \{0,1\}$ on $n = m' 2^{ck}$ inputs such that the monotone circuit depth of f is lower bounded by the (deterministic) communication cmplexity of $S(F) o g^n$.*

Proof

We start by defining f. Each of its n input coordinates will be indexed by a pair (C, l) where C is a constraint of F and l is a labelling of the Alice-variables in C. (That is, for each variable z_i in C, l specifies a value from X.)

For a given assignment to the inputs of f (an assignment to the variables $v_{C,l}$), f is 1 if and only if there is a global labelling l of all of the underlying variables z_i such that for each constraint C, $v_{C,l|vars(C)} = 1$.

This function is clearly monotone. Now we want to show how to reduce the search problem $S(F) o g^m$ to the monotone KW-game for f. To this end, let (x, y) be an input to the search problem

$S(f)og^m$. Alice (using x) computes an assignment $\alpha \in f^{-1}(1)$ to the underlying variables of f as follows:

Given her assignment to all of the X-variables, she sets the corresponding variables $v_{C,l}$ to true.

Bob (using y) computes an assignment $\beta \in f^{-1}(0)$ to the underlying varibles of f as follows: Variable $v_{C,l}$ is set to 1 by Bob if and only if under the assignment given by l to Alice's half of the variables underlying C, and under the assignment given by y to Bob's half of the variables underlying C the constraint C is satisfied. Since F is unsatisfiable, there is no global labelling satisfying all of the constraints, so y will be a 0-input of f.

Now Alice and Bob run a protocol for the KW-game on x and y described above. The output of the protocol will be some variable $v_{C,l}$ that is 1 in Alice's instance but 0 in Bob's instance. Because Alice's 1 instance was constructed so that for each constraint c exactly one coordinate of the form $(C, \cdot)$ is 1, we must have that l is the restriction of x to the variables of C. On the other hand, Bob's construction of the 0 instance tells us that C is not satisfied when Alice's half of the assignment comes from x and Bob's half of the assignment comes from y. Thus we have found a violated constraint C for the original search problem.

Intuitively, a 1-assignment for f is specified by a value to all of Alice's half of the variables x: she sets $v_{C,l}$ to true if the labelling l is the value specified by her half of the variables. A 0-assignment for f is specified by a value for all of Bob's half of the variables y: He looks at each variable $v_{C,l}$ and sets it to 1 if under the assignment specified by l to Alice's half of the variables, and using y for Bob's half, makes C true. We know that there is no x, y pair that will satisfy all clauses so there must be some clause that is violated, and therefore there is no global assignment – hence Bob's assignment to the $v_{C,l}$ variables will be a 0-assignment.

Solving the KW search problem for f thus finds a variable $v_{C,l}$ that they give different values to, and this corresponds to a constraint C that was falsified by their original assignment x, y.