IAS/Park City Mathematics Series
Volume 3, 1993

Circuit and Communication Complexity

Ran Raz

LECTURE 6
Communication complexity and circuit depth

There is an interesting connection between communication complexity and the depth of Boolean circuits. In the next lectures, we will explore this connection and we will use it to prove lower bounds for the depth of monotone Boolean circuits. In particular, we will prove a lower bound of $\Omega(\log^2 n)$ for the monotone depth of directed st-Connectivity and a lower bound of $\Omega(n)$ for the monotone depth of Matching.

6.1. Karchmer - Wigderson Games

For any function $f : \{0,1\}^n \to \{0,1\}$, we define a corresponding communication game G_f. The game G_f is referred to as, the KW game corresponding to the function f.

Definition 1. *(KW Game, G_f):* Given $f : \{0,1\}^n \to \{0,1\}$, define the communication game G_f as follows:

- Player 1 gets $x \in \{0,1\}^n$, s.t., $f(x) = 1$.
- Player 2 gets $y \in \{0,1\}^n$, s.t., $f(y) = 0$.

The goal is to find a coordinate i, s.t., $x_i \neq y_i$.

Since $x \neq y$, there is at least one coordinate i, such that, $x_i \neq y_i$. That is, there is at least one right answer for the game G_f on input (x, y). Note however, that in many cases there is more than one such coordinate i. Thus, the game G_f is

[1] Faculty of Mathematics and Computer Science, The Weizmann Institute of Science, Rehovot 76100, Israel.
E-mail address: ranraz@wisdom.weizmann.ac.il.

November 15, 2000

Based on lecture notes taken by Yael Tauman, Yoav Rodeh and Yoad Lustig.

a relation, rather than a function. That is, for an input (x, y) there may be more than one right answer. This is different than communication problems that were discussed in previous lectures. Nevertheless, we can still define the communication complexity of the game as before. We denote the deterministic communication complexity of a game G by $CC(G)$.

Why is the game G_f interesting ? It turns out that the deterministic communication complexity of the game G_f is exactly equal to the circuit depth of the function f (i.e., the minimal depth of a Boolean circuit that computes f). For a Boolean circuit C, denote its depth by $Depth(C)$. For a function f, denote its circuit depth by $Depth(f)$.

Lemma 6.1.1. For every $f : \{0,1\}^n \to \{0,1\}$,

$$CC(G_f) = Depth(f).$$

Proof. We will first show that $CC(G_f) \leq Depth(f)$. Given a Boolean circuit C that computes f, we will construct a communication protocol for the game G_f, with communication complexity $Depth(C)$. The proof is by induction on $Depth(C)$.

Base case: $Depth(C) = 0$. In this case, $f(z)$ is simply the function z_i or $\neg z_i$, for some i. Therefore, there is no need for communication, since i is a coordinate in which x and y always differ. That is, the two players can give the answer i, for any input pair (x, y). This is a protocol for G_f, with communication complexity 0.

Induction step: Consider the top gate of C. Assume first that $C = C_1 \wedge C_2$. Then,

$$Depth(C_1), Depth(C_2) \leq Depth(C) - 1.$$

Denote by f_1 and f_2 the functions computed by C_1 and C_2 respectively. By the inductive hypothesis,

$$CC(G_{f_1}), CC(G_{f_2}) \leq Depth(C) - 1.$$

We know that $f(x) = 1$ and $f(y) = 0$. Therefore,

$$\begin{array}{c} f_1(x) = f_2(x) = 1 \\ f_1(y) = 0 \ \text{ or } \ f_2(y) = 0 \end{array}$$

Let us describe the protocol for G_f. In the first step of the protocol, Player 2 sends a value in $\{1, 2\}$, indicating which of the functions f_1 or f_2 is zero on y. Assume that Player 2 sends 1. In this case, both players know:

$$\begin{array}{c} f_1(y) = 0 \\ f_1(x) = 1 \end{array}$$

Hence, to solve the game G_f, the players can apply a protocol for G_{f_1}. By the inductive hypothesis, there is such a protocol with communication complexity $CC(G_{f_1}) \leq Depth(C) - 1$. In the same way, if Player 2 sends 2 the players can use the protocol for G_{f_2}. The players used only one additional bit of communication. Hence, we can conclude that

$$\begin{array}{rcllcl} CC(G_f) & \leq & 1 + \max\{CC(G_{f_1}), CC(G_{f_2})\} & & & \\ & \leq & 1 + (Depth(C) - 1) & & \leq & Depth(f). \end{array}$$

We assumed that $C = C_1 \wedge C_2$. The other case, $C = C_1 \vee C_2$, is proved in the same way, except that Player 1 is the one who sends the first bit (indicating whether $f_1(x) = 1$ or $f_2(x) = 1$).

We will now show that $CC(G_f) \geq Depth(f)$. To prove this, we define a more general communication game. For any two disjoint sets: $A, B \subseteq \{0,1\}^n$, denote by $G_{A,B}$ the following game:

- Player 1 gets $x \in A$.
- Player 2 gets $y \in B$.
- The goal is to find a coordinate i, s.t., $x_i \neq y_i$.

Note that G_f is the same as $G_{f^{-1}(1),f^{-1}(0)}$. We will prove the following claim:

Claim 6.1.2. If $CC(G_{A,B}) = d$ then there is a function $f : \{0,1\}^n \to \{0,1\}$, such that:

- $f(x) = 1$, for every $x \in A$.
- $f(y) = 0$, for every $y \in B$.
- $Depth(f) \leq d$.

That is, the function f separates A from B, and $Depth(f) \leq d$. Note that for the game $G_f = G_{f^{-1}(1),f^{-1}(0)}$, a function that separates A from B must be the function f itself. Hence, we obtain that $Depth(f) \leq CC(G_f)$, as required. Let us give the proof of the claim.

Proof. *(claim)* The proof is by induction on $d = CC(G_{A,B})$.

Base case: $d = 0$. That is, the two players know the answer without any communication. Hence, there is a coordinate i, such that, for every $x \in A$ and every $y \in B$, we have $x_i \neq y_i$. Thus, the function $f(z) = z_i$ or the function $f(z) = \neg z_i$ satisfies the requirements of the claim (depending on whether for every $x \in A$ we have $x_i = 1$, or, for every $x \in A$ we have $x_i = 0$).

Induction step: We have a protocol of communication complexity d for the game $G_{A,B}$. Assume first that Player 1 sends the first bit in the protocol. This bit partitions the set A into two disjoint sets $A = A_0 \cup A_1$. If the first bit is 0, the rest of the protocol is a protocol for the game $G_{A_0,B}$. If the first bit is 1, the rest of the protocol is a protocol for the game $G_{A_1,B}$. Hence, for both games, $G_{A_0,B}$ and $G_{A_1,B}$, we have protocols with communication complexity at most $d-1$. By the inductive hypothesis, we have two functions f_0 and f_1 that satisfy:

- $f_0(x) = 1$, for every $x \in A_0$.
- $f_1(x) = 1$, for every $x \in A_1$.
- $f_0(y) = f_1(y) = 0$, for every $y \in B$.
- $Depth(f_0), Depth(f_1) \leq d-1$.

We define $f = f_0 \vee f_1$. Then:

- For every $x \in A$, we have $f(x) = f_0(x) \vee f_1(x) = 1$.
- For every $y \in B$, we have $f(y) = f_0(y) \vee f_1(y) = 0$
- $Depth(f) = 1 + \max\{Depth(f_0), Depth(f_1)\} \leq d$

That is, f satisfies the requirements.

If Player 2 sends the first bit, B is partitioned into two disjoint sets, $B = B_0 \cup B_1$, and as before, the rest of the protocol is a protocol for the games G_{A,B_0} and G_{A,B_1}, (depending on the bit sent). By the inductive hypothesis, we have two functions, g_0, g_1, corresponding to the two games, G_{A,B_0} and G_{A,B_1}, such that,

- $g_0(x) = g_1(x) = 1$, for every $x \in A$.
- $g_0(y) = 0$, for every $y \in B_0$.
- $g_1(y) = 0$, for every $y \in B_1$.

We define $g = g_0 \wedge g_1$. Then:

- For every $x \in A$, we have $g(x) = g_0(x) \wedge g_1(x) = 1$.
- For every $y \in B$, we have $g(y) = g_0(y) \wedge g_1(y) = 0$.
- $Depth(g) = 1 + \max\{Depth(g_0), Depth(g_1)\} \leq d$.

■

■

Consider for example the following game. Player 1 gets a graph x (on n vertices) that contains a clique of size $n/2$. Player 2 gets a graph y (on n vertices) that doesn't contain a clique of size $n/2$. The goal of the two players is to find an edge in x that doesn't exist in y or an edge in y that doesn't exist in x. Lemma 6.1.1 shows that the communication complexity of this game is exactly equal to the circuit depth of the $(n/2)$-Clique function. In particular, one can try to prove a lower bound for the circuit depth of the $(n/2)$-Clique function, by proving a lower bound for the communication game. Note that no lower bound better than $\Omega(\log n)$ was ever proved for the circuit depth of an explicit Boolean function. KW games give a direction to try to prove such lower bounds.

6.2. Monotone Complexity

A monotone Boolean function is defined in the following way.

Definition 2. *(Monotone Function):* $f : \{0,1\}^n \to \{0,1\}$ is a *monotone function* if for every $x, y \in \{0,1\}^n$, $x \geq y$ implies $f(x) \geq f(y)$, where the partial order $\geq$ on $\{0,1\}^n$ is the Hamming order, that is, $(x_1, \ldots, x_n) \geq (y_1, \ldots, y_n)$ iff for every $1 \leq i \leq n$ we have $x_i \geq y_i$.

We think of the Hamming partial order also as the containment order between sets, where a vector $x \in \{0,1\}^n$ corresponds to the set $S_x = \{i \mid x_i = 1\}$. Obviously, $x \leq y$ iff $S_x \subseteq S_y$.

Many well studied functions on graphs (e.g., k-Clique, Perfect-Matching, st-Connectivity, etc.) are monotone functions. One example is the k-Clique function,

$$CLIQUE_{n,k} : \{0,1\}^{\binom{n}{2}} \to \{0,1\}.$$

The domain of $CLIQUE_{n,k}$ is the set of all graphs on the n vertices $\{1, \ldots n\}$. A graph is represented by an assignments to the $\binom{n}{2}$ variables $x_{i,j}$, where for every pair $i, j \in \{1, \ldots n\}$, $x_{i,j} = 1$ iff (i,j) is an edge in the graph. $CLIQUE_{n,k}$ gets the value 1 on a graph iff the graph contains a clique of size k.

Another example that will be analyzed in the next lectures is the directed st-Connectivity function.

Definition 3. *(Directed st-Connectivity):* Given a directed graph G on $n+2$ vertices, two of which are marked as s and t, *st-Connectivity*$(G) = 1$ iff there is a directed path from s to t in G.

Obviously, *st-Connectivity* is a monotone function, since if we add edges we cannot disconnect an existing path from s to t.

Every monotone function can be characterized by the set of its minterms and by the set of its maxterms.

Definition 4. *(Minterm, Maxterm):* Let $f : \{0,1\}^n \rightarrow \{0,1\}$ be a monotone function.

- A *minterm* of f is $x \in \{0,1\}^n$, s.t., $f(x) = 1$ and for every $x' < x$, $f(x') = 0$.
- A *maxterm* of f is $y \in \{0,1\}^n$, s.t., $f(y) = 0$ and for every $y' > y$, $f(y') = 1$.

For example, for *st-Connectivity*: A minterm is a graph that contains one simple directed path from s to t (i.e., a path from s to t that does not intersect itself), and no other edges.

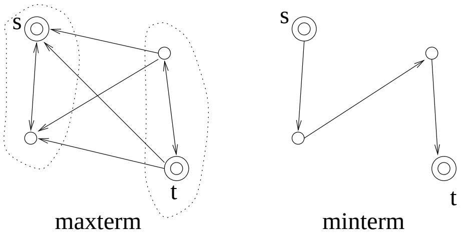

Figure 1. A maxterm and a minterm of *st-Connectivity*

- If a graph G is a minterm then it must contain a simple path P from s to t. G cannot contain any other edges, because then $P < G$ (in the edge containment order), in contradiction to the fact that G is a minterm.
- Every simple path from s to t is a minterm, because every edge that we drop from it disconnects s from t.

A maxterm for *st-Connectivity* is a graph G, such that, the set of vertices can be partitioned into two disjoint sets S and T, such that,

- $s \in S$ and $t \in T$.
- G contains all possible directed edges, except those from S to T.

This is indeed the set of maxterms for *st-Connectivity*, because

- If G is a maxterm then let S be the set of vertices that are reachable from s in G, and let T be the set of all other vertices. $t \in T$, because one cannot reach t from s in G, since *st-Connectivity*$(G) = 0$. G contains all edges except those from S to T, otherwise we can add the missing edges and still leave t unreachable from s. There are no edges from S to T by the definition of S (recall that S is the set of vertices reachable from s).
- If G satisfies both criteria, then every path starting from s in G will remain in S and therefore will not reach t. Thus, *st-Connectivity*$(G) = 0$. Every edge we add to G will connect S to T, and since S and T are strongly connected it will create a path from s to t.

We think of a maxterm of *st-Connectivity* as a partition of the set of vertices into two sets (S and T) or as a two coloring of the vertices by the colors 0 and 1 (where S is colored by 0 and T is colored by 1).

Monotone Boolean circuits are defined in the same way as Boolean circuits, except that we do not allow to use *NOT* gates. Intuitively, monotone circuits cannot compute all functions, because there is no way to simulate a *NOT* gate using *AND* and *OR* gates. Nevertheless, it is not hard to see that monotone circuits can compute all monotone functions. This can be done, e.g., by taking the disjunction (of clauses that correspond to) all the minterms or the conjunction of (clauses that correspond to) all the maxterms.

Proposition 6.2.1. A function $f : \{0,1\}^n \to \{0,1\}$ is monotone iff it can be computed by a monotone circuit.

The monotone circuit size and the monotone circuit depth of a monotone Boolean function are defined in a similar way to the definitions of the circuit size and the circuit depth of a Boolean function.

Definition 5. *(Mon-Size, Mon-Depth):* For a monotone function $f : \{0,1\}^n \to \{0,1\}$, define:

(1) $\textit{Mon-Size}(f) \overset{\text{def}}{=}$ The minimal size of a monotone circuit for f.
(2) $\textit{Mon-Depth}(f) \overset{\text{def}}{=}$ The minimal depth of a monotone circuit for f.

Obviously, for every monotone function f,

- $\textit{Mon-Size}(f) \geq \textit{Size}(f)$.
- $\textit{Mon-Depth}(f) \geq \textit{Depth}(f)$.

There are functions for which these inequalities are strict.

Unlike for general Boolean circuits, several (famous) lower bounds were proved for the monotone size and for the monotone depth of certain functions. In particular, it was proved by Razborov that *mon-P* $\neq$ *mon-NP* and by Karchmer and Wigderson that *mon-*$NC_1 \neq$ *mon-*NC_2. For the k-Clique function, for example, it is known that (for certain values of k, depending on n):

$$\begin{aligned} \textit{Mon-Size}(CLIQUE_{n,k}) &= \Omega(2^{n^{1/3}/\log n}). \\ \textit{Mon-Depth}(CLIQUE_{n,k}) &= \Omega(n). \end{aligned}$$

Recall that for the non-monotone case, there are no non-trivial lower bounds. In particular, there is no lower bound better than linear for the circuit size of any explicit function, and there is no lower bound better than logarithmic for the circuit depth of any explicit function.

In these lectures, we will prove several lower bounds for the depth of monotone circuits. In particular, we will prove an $\Omega(n)$ lower bound for the monotone depth of the Matching function, and an $\Omega(\log^2 n)$ lower bound for the monotone depth of directed *st*-Connectivity. To prove these bounds, we will use the monotone version of KW games.

6.3. Monotone Karchmer-Wigderson Games

For any monotone function $f : \{0,1\}^n \to \{0,1\}$, we define a corresponding communication game M_f. The game M_f is referred to as, the monotone KW game corresponding to the function f.

Definition 6. *(Monotone KW Game, M_f):* Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, define the communication game M_f as follows.

- Player 1 gets $x \in \{0,1\}^n$, s.t., $f(x) = 1$.
- Player 2 gets $y \in \{0,1\}^n$, s.t., $f(y) = 0$.

The goal is to find a coordinate i, s.t., $x_i > y_i$, i.e., $x_i = 1$ and $y_i = 0$.

The game is similar to the game G_f, except that f is monotone, and that the goal is more specific. The goal in M_f is to a find a coordinate i, such that, $x_i > y_i$, rather than $x_i \neq y_i$ in the game G_f. Note that the goal is always achievable. Otherwise, for every i we have $y_i \geq x_i$, and hence $y \geq x$, in contradiction to the fact that f is monotone and $f(x) = 1$, $f(y) = 0$.

The deterministic communication complexity of the game M_f is exactly equal to the monotone circuit depth of the function f.

Lemma 6.3.1. For every monotone $f : \{0,1\}^n \to \{0,1\}$,

$$CC(M_f) = Mon\text{-}Depth(f).$$

Proof. The proof is similar to the proof for the non-monotone case, with the following minor modifications.

When constructing the protocol from a given circuit:

Base case: since f is monotone, if the depth is 0, we have that $f(z) = z_i$ and therefore it must be the case that $x_i = 1$ and $y_i = 0$. Hence, $x_i > y_i$, and as before, there is no need for communication, as the answer is always i.

Induction step: In the induction step, the top gate separates our circuit into two sub-circuits. The protocol then uses one communication bit to decide which of the two games (corresponding to the two sub-circuits) to solve. Since the sub-circuits are monotone, by the inductive hypothesis they each have a protocol that solves their corresponding monotone game. This solves the monotone game corresponding to the original circuit, since the sub-games are monotone, and therefore the coordinate i that is found satisfies $x_i > y_i$.

When constructing the circuit from a given protocol:

Base case: if there is no communication, both players already know a coordinate i, such that, $x_i > y_i$. Hence, our circuit is simply $f(z) = z_i$.

Induction step: Each communication bit splits our game into two sub-games of smaller communication complexity. Note that if the original game was a monotone game (i.e., if the goal is to find i, s.t., $x_i > y_i$), so are the two sub-games. By the inductive hypothesis, the circuits for these games are monotone. Since we only added either an AND gate or an OR gate, the entire circuit constructed is also monotone.

■

Consider for example the following game. Player 1 gets a graph x (on n vertices) that contains a clique of size $n/2$. Player 2 gets a graph y (on n vertices) that doesn't contain a clique of size $n/2$. The goal of the two players is to find an edge in x that doesn't exist in y. Lemma 6.3.1 shows that the communication complexity of this game is exactly equal to the monotone circuit depth of the $(n/2)$-clique function. Using this connection, a lower bound of $\Omega(n)$ was proved for the monotone depth of the $(n/2)$-Clique function.

We will now define the communication game $\hat{M}_f$. The game $\hat{M}_f$ is equivalent to M_f and will usually be more convenient to work with.

Definition 7. *(Monotone KW Game, $\hat{M}_f$):* Given a monotone function $f : \{0,1\}^n \to \{0,1\}$, define the communication game $\hat{M}_f$ as follows:

- Player 1 gets $x \in \{0,1\}^n$, s.t., x is a minterm of f (hence, $f(x) = 1$).
- Player 2 gets $y \in \{0,1\}^n$, s.t., y is a maxterm of f (hence, $f(y) = 0$).

The goal is to find a coordinate i, s.t., $x_i > y_i$, i.e., $x_i = 1$ and $y_i = 0$.

Note that the game $\hat{M}_f$ is a restriction of the game M_f to a subset of inputs. Hence, any protocol for M_f is also a protocol for $\hat{M}_f$. Therefore,

$$CC(\hat{M}_f) \leq CC(M_f).$$

Actually, the communication complexity of the two games is exactly the same.

Proposition 6.3.2. For every monotone $f : \{0,1\}^n \rightarrow \{0,1\}$,

$$CC(\hat{M}_f) = CC(M_f).$$

Proof. We have to prove that $CC(\hat{M}_f) \geq CC(M_f)$. Given a protocol for $\hat{M}_f$, we will construct a protocol for M_f (with the same communication complexity).

Let x, y be inputs for the game M_f. Player 1 gets x, s.t., $f(x) = 1$, and finds a minimal x', s.t., $x' \leq x$ and $f(x') = 1$. This is done by successively changing coordinates in x from 1 to 0, while checking that $f(x')$ is still 1. Eventually, Player 1 has a minterm $x' \leq x$. In the same way, Player 2 finds a maxterm $y' \geq y$. The players apply the protocol for $\hat{M}_f$ on the input (x', y'). Since x' is a minterm, and y' is a maxterm, the protocol outputs a coordinate i, such that, ${x'}_i = 1$ and ${y'}_i = 0$. Since $x' \leq x$ and $y' \geq y$, we have $x_i = 1$ and $y_i = 0$. ■

As a corollary, we obtain,

Lemma 6.3.3. For any monotone function $f : \{0,1\}^n \rightarrow \{0,1\}$,

$$\textit{Mon-Depth}(f) = CC(\hat{M}_f).$$

Consider for example the following game. Player 1 gets a directed path x (on $n+2$ vertices) that starts from a vertex s and ends at a vertex t. Player 2 gets a coloring of the $n+2$ vertices by the colors $0, 1$, such that, s is colored 0 and t is colored 1. The goal of the two players is to find an edge (u, v) in x, such that, u is colored 0 and v is colored 1. Lemma 6.3.3 shows that the communication complexity of this game is exactly equal to the monotone circuit depth of directed *st-Connectivity*. A simple protocol for this game is the following: In the first round, Player 1 sends the name (i.e., number) of the middle vertex in the path x, and Player 2 replies with its color (according to the coloring y). If the color of the middle vertex is 0 then the players continue with the second half of the path, and if the color is 1 then the players continue with the first half of the path. The players continue to perform a binary search, until they are left with a path of length 1. This path will be an edge (u, v), such that, u is colored 0 and v is colored 1. In each round of the protocol, the players communicate $O(\log n)$ bits (the name of the vertex and its color). Since in each step the path is shorten by half, the number of rounds will be $O(\log n)$. Altogether, the communication complexity of the protocol is $O(\log^2 n)$. Hence, by Lemma 6.3.3, the monotone circuit depth of directed *st-Connectivity* is $O(\log^2 n)$. In the next lectures, we will show that this upper bound is tight. That is, we will prove a lower bound of $\Omega(\log^2 n)$ for the communication complexity of the game, and hence also for the monotone circuit depth of directed *st-Connectivity*. This shows that $\textit{mon-NC}_1 \neq \textit{mon-NC}_2$.

6.4. Lower Bound for Matching

We will now prove a lower bound of $\Omega(n)$ for the monotone depth of the Matching function. For simplicity, we define the function $Match$ as follows. Let $n = 3k$. The function $Match$ inputs a graph on n vertices and outputs 1 if the graph contains k independent edges (i.e., a k-matching), and outputs 0 otherwise.

Denote the corresponding monotone KW game by M_0. The game M_0 is the following. Player 1 gets a graph x (on n vertices) that contains a k-matching. Player 2 gets a graph y (on n vertices) that doesn't contain a k-matching. The goal of the two players is to find an edge in x that doesn't exist in y. By Lemma 6.3.1,

$$CC(M_0) = Mon\text{-}Depth(Match).$$

Hence, in order to prove a lower bound for $Mon\text{-}Depth(Match)$, it is enough to prove a lower bound for $CC(M_0)$.

Consider the following game, denoted by M_1. Player 1 gets a k-matching x (on n vertices), that is, k independent edges. Player 2 gets a set y of $k-1$ vertices. The goal is to find an edge in x that does not touch any of the vertices in y.

Claim 6.4.1.

$$CC(M_1) \leq CC(M_0).$$

Proof. Note that every k-matching x is a minterm of the function $Match$. Every set y of $k-1$ vertices can be viewed as a maxterm of the function $Match$, by considering all the edges that touch y. Hence, any protocol P for M_0 can be applied on (x, y) to get an edge in x that doesn't touch y. That is, any protocol P for M_0 can be applied also as a protocol for M_1. ■

Thus,

$$CC(M_1) \leq Mon\text{-}Depth(Match).$$

Hence, to prove a lower bound for $Mon\text{-}Depth(Match)$, it is enough to prove a lower bound for $CC(M_1)$. Let P be a protocol for M_1. Observe that we can assume that P outputs each possible right answer with the exact same probability (i.e., if for the input (x, y) there are several right answers then the protocol outputs each one of them with the same probability). This can be assumed, since (using the common random string) the players can randomly permute the vertices before applying the protocol P.

Consider now the following game, denoted by M_2. Player 1 gets a k-matching x. Player 2 gets a set y of k vertices. The goal of the two players is to output 1 if there is an edge in x that does not touch any of the vertices in y, and output 0 if every edge in x touches a vertex in y.

Claim 6.4.2. For any constant $\epsilon > 0$, there exists a constant a, such that,

$$CC_\epsilon(M_2) \leq a \cdot CC(M_1).$$

Proof. Assume that we have a deterministic protocol P_1 for M_1. We will construct a probabilistic protocol P_2 for M_2 (with the same communication complexity as P_1). Let x, y be inputs for the game M_2. Player 2 has a set y of k vertices, and will randomly choose a vertex $v \in y$ and remove it. Now, Player 2 is left with a set y' of $k-1$ vertices. The two players can now apply the protocol P_1 (for M_1) on the input (x, y') and obtain (as an output) an edge $e \in x$ that doesn't touch any of the vertices in y'. If the edge e does not touch v then the protocol P_2 (for M_2) outputs

1 (as e is an edge that doesn't touch any vertex in y). If e touches v the protocol P_2 outputs 0 (i.e., P_2 assumes that there is no edge in x that doesn't touch y, as such an edge was not found by P_1).

The analysis of the protocol P_2 is simple. If P_2 outputs 1 then there can be no mistake (e does not touch neither v nor any of the other vertices in y, as assured by the protocol P_1). On the other hand, if the protocol outputs 0 then a mistake is possible. It may be that there is an edge e' in x that doesn't touch any of the vertices in y, but still the protocol P_1 outputs the edge e that does touch v. However, since the edges are independent, there is at most one edge e that touches v, and we are analyzing the case where there is at least one edge $e' \in x$ that doesn't touch any of the vertices in y. As mentioned above, we can assume that the protocol P_1 outputs each possible right answer with the exact same probability. An error occurs if e was output (and not any of the edges e'). Hence, the probability of error is at most 1/2 (it may be smaller if there are several edges e' that do not touch any vertex in y).

To reduce the probability of error to any constant ϵ, one can repeat the protocol a constant number of times. ■

Thus,

$$\Omega(CC_\epsilon(M_2)) \leq Mon\text{-}Depth(Match).$$

We will prove that

$$CC_\epsilon(M_2) = \Omega(n),$$

by a reduction from the 3-Distinctness problem.

Recall that in the 3-Distinctness problem the inputs are $x, y \in \{a, b, c\}^n$. To convert their input to an input for M_2, the players construct the following graph. For each coordinate construct an independent triangle, and denote the triangle vertices by a, b, c. Denote each edge of a triangle by the same letter as the vertex that it does not touch (see Figure 2). The players convert their inputs to inputs for M_2 in the following way: Player 1 interprets his n coordinates as the corresponding n edges in the n triangles (one edge for each coordinate). That is, each x_i is interpreted as the corresponding edge in the i^{th} triangle. Denote the set of these edges by $\hat{x}$. Player 2 interprets his n coordinates as the corresponding n vertices in the n triangles. That is, each y_i is interpreted as the corresponding vertex in the i^{th} triangle. Denote the set of these vertices by $\hat{y}$. Obviously, there is an edge in $\hat{x}$ that doesn't touch $\hat{y}$ iff there is a coordinate i, such that, $x_i = y_i$.

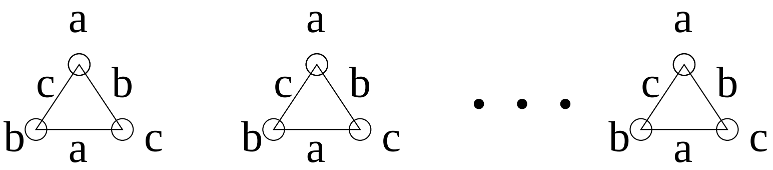

Figure 2. Reducing 3- Distinctness to M_2

Recall that we have a lower bound of $\Omega(n)$ for the probabilistic communication complexity of the 3-Distinctness problem. Hence, we obtain a lower bound of $\Omega(n)$ for the probabilistic communication complexity of M_2. Hence,

$$Mon\text{-}Depth(Match) = \Omega(n).$$