CSG399: Gems of Theoretical Computer Science. Lectures 25-27. Apr. 10-21, 2009.
Instructor: Emanuele Viola Scribe: Fangfei Zhou

Communication Complexity

This lecture deals with Communication Complexity, which is the study of the amount of information that needs to be exchanged among two or more parties (or players) which are interested in reaching a common computational goal.

1 Two parties

We start with the model in which there are only 2 parties, A and B. They have the following properties:

- They collaborate, and
- each party has unlimited computing power.

Their task is to compute a predefined function of two inputs

$$f : X \times Y \to \{0, 1\}$$

where A only knows $x \in X$, and B only knows $y \in Y$. The parties A and B engage in a communication protocol and exchange bits. At each step, the protocol specifies whose turn is to speak, or if the protocol is over. This is a function of the bits exchanged so far. If a party is to speak, the protocol specifies which bit is sent, and this is a function of both the bits exchanged so far and the input to the party who is to speak. If the protocol is over, the last bit exchanged is the output. We say that the protocol uses c bits if for every input A and B exchange $\leq c$ bits. The bits exchanged are called *transcript*.

We can visualize a protocol via a binary tree (Figure 1). Each node is labeled with a party and a function from that party's input to $\{0, 1\}$, which specifies which children to go to.

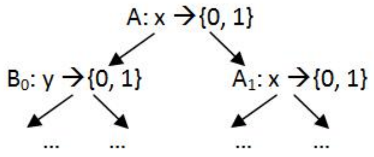

Figure 1: Protocol tree

1.1 The communication complexity of equality

Consider the function Equality : $\{0,1\}^n \times \{0,1\}^n \to \{0,1\}$, Equality$(x,y) = 1 \Leftrightarrow x = y$. Trivially, Equality can be computed with communication $n+1$: A sends her input to B; B then communicates the value of Equality. The same trivial upper bound holds $\forall f : \{0,1\}^n \times \{0,1\}^n \to \{0,1\}$. We now prove the following lower bound.

Theorem 1. *Any protocol for equality must exchange at least n bits.*

Before proving this theorem, we cover some properties of protocols.

Definition 2. *A rectangle in $X \times Y$ is a subset $R \subseteq X \times Y$ such that $R = A \times B$ for some $A \subseteq X$ and $B \subseteq Y$.*

An equivalent definition is given by the following proposition.

Proposition 3. *$R \subseteq X \times Y$ is a rectangle iff $(x,y) \in R$ and $(x',y') \in R \Rightarrow (x,y'),(x',y) \in R$.*

Lemma 4. *Let P be a protocol that uses c bits, let $t \in \{0,1\}^c$. The set of inputs that induce communication t is a rectangle.*

Proof. Let $A \subseteq X \times Y$ be the set of inputs that induce communication t. Suppose that $(x,y),(x',y') \in A$, we want to show that $(x,y') \in A$ (similarly for (x',y)). We prove by induction on i that the i-th bit exchanged by P on input (x,y') is t_i. Of course this means that the protocol exchanges t on input (x,y') and so $(x,y') \in A$ as desired.

For $i = 1$, the bit sent by A only depends on x, but we know $P(x,y)$ exchanges t_1, so we are done.

For general i, suppose it is A's turn to speak. The bit she sends is a function of x and the communication so far. By induction hypothesis the communication so far is $t_1, \ldots, t_{i-1}$. So A cannot distinguish between (x,y) and (x,y') and will send t_i as next bit.

If it is B's turn to speak, we reason in the same way replacing (x,y') with (x',y'). □

Corollary 5. *Suppose $f : X \times Y \to \{0,1\}$ is computable by a c-bit protocol, then there is a partition of $X \times Y$ in 2^c rectangles, where each rectangle is f-monochromatic (i.e., all inputs in the rectangle give the same value of f).*

Proof. For each transcript $t \in \{0,1\}^c$, consider $R_t :=$ the set of inputs that induce t. R_t is a rectangle by the previous lemma. It is obviously a partition and f-monochromatic. □

Figure 2 shows two ways to partition equality in monochromatic rectangles. We can now prove the lower bound for equality.

Proof that Equality requires communication n. Assume we can partition $X \times Y$ in equality-monochromatic rectangles. Consider the 2^n inputs (e,e) where $e \in \{0,1\}^n$. Observe that no equality-monochromatic rectangle can contain both (e,e) and (b,b) if $e \neq b$, for else (e,b) is in the rectangle, but since $e \neq b$ this cannot be equality-monochromatic.

Since the rectangles must cover all of the 2^n inputs (e,e), we need $\geq 2^n$ rectangles which implies that any protocol must use at least n bits of communication. □

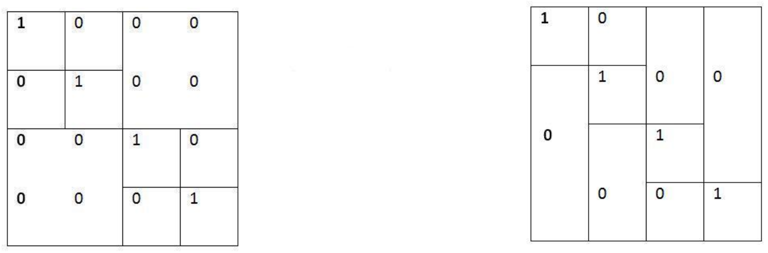

Figure 2: Two ways to partition equality in monochromatic rectangles.

1.2 Correlation bounds

Having established lower bounds, we now turn to correlation bounds. Specifically we seek explicit functions $f : X \times Y \to \{0, 1\}$ such that

$$\mathrm{Cor}(f, c - \mathrm{bit}) \leq \varepsilon$$

where recall the left-hand side above equals the maximum over all c-bit protocols P of

$$|E_{x,y} e[f(x, y) + P(x, y)]|$$

where $e(z) := (-1)^z$.

We will prove that for the inner production function

$$IP(x, y) := \Sigma_i x_i \cdot y_i \bmod 2$$

we have

$$\mathrm{Cor}(IP, c - \mathrm{bit}) \leq 2^c \cdot 2^{-\Omega(n)}.$$

In particular, for $c \ll n$, the correlation is $2^{-\Omega(n)}$.

Before proving this we present an application to Turing Machines lower bounds. A correlation bound is sufficient but not necessary for this application; for example the equality function (which being very biased has high correlation with 0-bit protocols) is sufficient but one needs to show a lower bound in a slightly different model than what studied before.

1.3 Application to Turing Machines lower bounds

Consider a single-tape read/write Turing Machine.

Theorem 6. *Suppose* $f : \{0, 1\}^n \times \{0, 1\}^n \to \{0, 1\}$ *satisfies* $\mathrm{Cor}(f, (n/500) - \mathrm{bit}) \leq 1/2$. *Then recognizing the language* $\{x0^n y : f(x, y) = 1\}$ *requires time* $\Omega(n^2)$ *on a single-tape Turing Machine.*

Challenge: Give an explicit language that requires time $\omega(n^2)$ on single-tape Turing Machines.

Proof sketch. Assume that a Turing Machine M with 2^q states decides the language in time $\epsilon \cdot n^2/q$ for small enough ϵ. We show a protocol that correlates with f.

For $i = 1, 2, ...n$, define the protocol P_i as follows: A is in charge of first $n+i$ cells (which include x); B is in charge of last $n + (n - i)$ cells (which include y). They simulate the Turing Machine in turn, communicating $2q$ bits whenever M crosses the boundary of the $(n + i)$-th cell. These bits represent the state of the machine or a special symbol denoting that the computation is over with final state s, from which the value of the function can be determined. The parties carry this simulation for up to $n/(1000q)$ crossings, after which they do as if the machine stopped in, say, state 0. The total amount of communication is $2qn/(1000q) = n/500$.

For a sufficiently small ϵ, it can be shown that there exists a P_i that has correlation at least 1/2 with f (exercise). $\square$

2 k parties, number-on-forehead

There are various ways in which we can generalize the 2-party model of communication complexity to $k > 2$ parties. The obvious generalization is to let k players compute a k-argument function $f(x_1, \ldots, x_k)$ where the i-th party only knows the i-th argument x_i. This model is useful in some scenarios, but we will focus on a different, fascinating model which has an unexpected variety of applications: the "Number on the Forehead" model. Here, again $f(x_1, \ldots, x_k)$ is a Boolean function whose input is k arguments, and there are k parties. The twist is that the i-th party knows all inputs except x_i, which we can imagine being placed on his forehead. Communication is broadcast.

Challenge: Give an explicit function $f : \overbrace{\{0,1\}^n \times ... \times \{0,1\}^n}^{k} \to \{0,1\}$ that cannot be computed with $k := 2\log n$ parties exchanging k bits.

2.1 An application to circuit lower bounds

Consider ACC^0 circuits: polynomial-size, unbounded fan-in circuits that in addition to the *OR*, *AND* and *NOT* gates allow MOD_m gates (that is, gates that for some fixed m output 1 iff the number of input bits that are 1 is divisible by m). When m is any composite, no lower bounds for these circuits are known. We now show how a sufficiently strong communication lower bound would yield such a circuit lower bound.

Theorem 7. *$\forall c, m, \exists d$ s.t. any function $f : \{0,1\}^n \to \{0,1\}$ that is computable by circuits of size n^c, depth c, with AND, OR, NOT, MOD m gates, can also be computed by a ($\log^d n$)-party protocol communicating ($\log^d n$) bits (for any partition of the input bits in k blocks $x_1, \ldots, x_k$).*

The theorem follows immediately from the next two lemmas.

Lemma 8. *The function f in Theorem 7 can be computed by a depth-2 circuit of size $2^{\log^{O(c)} n}$ where the output is a symmetric function (i.e., only depends on the number of bits that are 1 in the input to that gate) and the other gates are AND with fan-in $\leq \log^{O(c)} n$.*

We will not prove this lemma because its proof is not related to communication.

Lemma 9. *Let $f : \{0,1\}^n \to \{0,1\}$ be computable by a depth-2 circuit of size s where the output is a symmetric function and the other gates are AND with fan-in d. Then there exists a $(d+1)$-party protocol for f which communicates $O(d \log s)$ bits (under any partition of the n input bits in $x_1, \ldots, x_{d+1}$).*

Proof. Fix an arbitrary partition of the input in $x_1, \ldots, x_{d+1}$. All that the players need to compute is the number of AND gates that evaluate to 1, because the output is a symmetric function of these AND gates. Consider any AND gate. Since it depends on at most d variables, it does not depend on the bits in one of the blocks $x_1, \ldots, x_{d+1}$. Say it does not depend on x_j. Then the j-th party can compute this AND without communication. So let us partition the AND gates among the parties so that each party can compute the gates assigned to her without communication. Each party evaluates all the AND gates assigned to her and broadcasts the number $\leq s$ of these gates that evaluate to 1. This takes a total of $O(d \log s)$ bits. □

2.2 Correlation bound for generalized inner product

We consider the generalized inner product function $GIP : (\{0,1\}^n)^k \to \{0,1\}$:

$$GIP(x_1, \ldots, x_k) := \sum_{i=1}^{n} \bigwedge_{j=1}^{k} (x_j)_i \mod 2.$$

Theorem 10. $\mathrm{Cor}(GIP, c-\text{bit}\ \ k-\text{party}) \leq 2^c \cdot 2^{-\Omega(n/4^k)}$.

To prove the theorem we associate to any function a quantity $R(f) \in \mathbb{R}$ such that

1. $\mathrm{Cor}(f, c-\text{bit}) \leq 2^c \cdot R(f)^{1/2^k}$. Note that $R(f)$ only depends on f. (Theorem 11.)
2. $R(GIP) \leq 2^{-\Omega(n/2^k)}$. (Section 2.4.)

The combination of these two things proves Theorem 10.

Intuition for $R(f)$: Think of $k = 2$; we saw that any 2-party c-bit protocol partitions the inputs in 2^c f-monochromatic rectangles. How about we check how well f can be so partitioned? Instead of picking an arbitrary rectangle, let us pick one in which each side has length 2, and see how balanced the function is there. If a "good" partition exists, with somewhat high probability our little rectangle should fall in a monochromatic rectangle, and we should always get the same values of f. Otherwise, we should get mixed values of f.

Specifically, for $k = 2$,

$$R(f) := Ee_{\binom{x_1^0, x_2^0}{x_1^1, x_2^1}} [f(x_1^0, x_2^0) + f(x_1^0, x_2^1) + f(x_1^1, x_2^0) + f(x_1^1, x_2^1)] \in \mathbb{R}.$$

In general, for any k:

$$R(f) := \underset{\binom{x_1^0, \ldots, x_k^0,}{x_1^1, \ldots, x_k^1}}{Ee} \left[\sum_{\varepsilon_1, \ldots, \varepsilon_k \in \{0,1\}} f(x_1^{\varepsilon_1}, \ldots, x_k^{\varepsilon_k}) \right] \in \mathbb{R}.$$

The next theorem shows we can use $R(f)$ to bound from above the correlation of f with c-bit k-party protocols.

Theorem 11. $\forall f : X_1 \times ... \times X_k \to \{0,1\}$, $\mathrm{Cor}(f, c - \mathrm{bit}) \leq 2^c \cdot R(f)^{1/2^k}$.

2.3 Proof of Theorem 11

We prove this theorem via a sequence of lemmas.

Definition 12. *A function $g_i : X_1 \times \ldots \times X_k \to \{0,1\}$ is a* cylinder *in the i-th dimension if $\forall (x_1, \ldots, x_k)$ and x_i' we have $g_i(x_1, \ldots, x_{i-1}, x_i, x_{i+1}, \ldots, x_k) = g_i(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_k)$. A set $S \subseteq X_1 \times \ldots \times X_k$ is a* cylinder intersection *if $\exists$ cylinders $g_1, \ldots, g_k$ such that $S = \{x : \prod g_i(x) = 1\}$.*

Recall we saw that a 2-party protocol partitions the input in monochromatic rectangles. The following extension of this fact to k parties is via cylinder intersections.

Claim 1. *Any c-bit k-party protocol for $f : \overbrace{\{0,1\}^n \times ... \times \{0,1\}^n}^{k} \to \{0,1\}$ partitions the inputs in 2^c f-monochromatic cylinder intersections.*

Proof. Fix a transcript t, and consider the set A_t of inputs yielding that transcript. We claim that A_t is a cylinder intersection. To see this, consider the cylinder functions $g_i(x) = 1 \Leftrightarrow$ "From the point of view of the i-th party, x could yield transcript t" $\Leftrightarrow \exists x_i'$ such that $P(x_1, \ldots, x_{i-1}, x_i', x_{i+1}, \ldots, x_k)$ yields transcript t.

Obviously if x is in A_t then $g_i(x) = 1$ for all i.

To see the converse, take some input $x = (x_1, \ldots, x_k)$ such that $g_i(x) = 1$ for all i. This means that $\exists (x_1', \ldots, x_k')$ such that

$$\begin{gathered} (x_1', x_2, \ldots, x_k) \text{ yields } t; \\ (x_1, x_2', \ldots, x_k) \text{ yields } t; \\ \ldots \ldots \\ (x_1, x_2, \ldots, x_k') \text{ yields } t. \end{gathered}$$

We must show that x yields t as well, i.e. $x \in A_t$. This is argued by induction on the bits in t, using the same "copy and paste" argument that was used for $k = 2$. $\square$

Using the notion of cylinder intersections we can now relate an arbitrary protocol to a special class of protocols p^*. Each protocol p^* can be written as $p^*(x) = \sum g_i(x) \ mod \ 2$, where g_i is a cylinder in i-th dimension. This corresponds to each party sending just one bit independently of the others, and the output of the protocol being the XOR of the bits.

Lemma 13. $\text{Cor}(f, c - \text{bit}) \leq 2^c \cdot \text{Cor}(f, \text{protocols}^*)$.

Proof. We use a general trick to turn products $\overbrace{\prod_i g_i(x) = 1}^{cylinder\ \ intersection}$ into sums $\overbrace{\sum g_i(x) \ mod \ 2}^{p^*}$.

Fix any c-bit protocol, let $\{x : \prod_i g_i^1(x) = 1\}, \ldots, \{x : \prod_i g_i^{2^l}(x) = 1\}$ be the corresponding 2^l f-monochromatic cylinder intersections (by the previous claim). Observe that for a fixed x,

$$\underset{y_1,\ldots,y_k \in \{-1,1\}}{E}\left[(y_1)^{1+g_1(x)} \cdot (y_2)^{1+g_2(x)} \cdot \ldots \cdot (y_k)^{1+g_k(x)}\right] = \begin{cases} 1 & \text{if } \exists i : g_i(x) = 0 \\ 0 & \text{if } \forall i : g_i(x) = 1. \end{cases}$$

Therefore,

$$e(p(x)) = \sum_{i=1}^{2^c} r(i) \underset{y_1,\ldots,y_k \in \{-1,1\}}{E}\left[(y_1)^{1+g_1^i(x)} \cdot (y_2)^{1+g_2^i(x)} \cdot \ldots \cdot (y_k)^{1+g_k^i(x)}\right]$$

where $r(i) \in \{-1, 1\}$ is the value of the protocol on the i-th cylinder intersection. Note that $\forall x$ exactly one expectation will be 1, the one corresponding to the cylinder intersection where x lands. So we have:

$$\begin{aligned}
&Ee[f(x) + p(x)] \\
&= \underset{x}{E}[e(f(x)) \cdot e(p(x))] \\
&= \underset{x}{E}\left[e(f(x)) \cdot \textstyle\sum_{i=1}^{2^c} r(i) \underset{y_1,\ldots,y_k \in \{-1,1\}}{E}\left[(y_1)^{1+g_1^i(x)} \cdot (y_2)^{1+g_2^i(x)} \cdot \ldots \cdot (y_k)^{1+g_k^i(x)}\right]\right] \\
&= \textstyle\sum_{i=1}^{2^c} \underset{\binom{x,}{y_1,\ldots,y_k \in \{-1,1\}}}{E}\left[e(f(x)) \cdot r(i) \cdot (y_1)^{1+g_1^i(x)} \cdot (y_2)^{1+g_2^i(x)} \cdot \ldots \cdot (y_k)^{1+g_k^i(x)}\right] \\
&\leq 2^c \cdot \underset{\binom{x,}{y_1,\ldots,y_k \in \{-1,1\}}}{E}\left[e(f(x)) \cdot r(i) \cdot (y_1)^{1+g_1^{i*}(x)} \cdot (y_2)^{1+g_2^{i*}(x)} \cdot \ldots \cdot (y_k)^{1+g_k^{i*}(x)}\right],
\end{aligned}$$

where $i*$ is the value of i that gives the largest summand. Now fix $y_1, \ldots, y_k$ to maximize the expectation, and let $J \subseteq \{1, \ldots, k\}$ be the indices corresponding to $y_j = -1$, i.e., $j \in J \Rightarrow y_j = -1$. We continue:

$$\begin{aligned}
Ee[f(x) + p(x)] &\leq 2^c \cdot \underset{x}{E}\left[e(f(x)) \cdot \prod_{j \in J} (-1)^{1+g_j^{i*}(x)}\right] \\
&= 2^c \cdot \underset{x}{E}\left[e(f(x) + \sum_{j \in J} (1 + g_j^{i*}(x))\right] \\
&= 2^c Ee[f(x) + p^*(x)] \leq 2^c \cdot \text{Cor}(f, \text{protocol}^*)
\end{aligned}$$

where p^* is the protocol* that computes $\sum_{j \in J} (1 + g_j^{i*}(x))$ modulo 2. □

Lemma 14. *For every function $g := X_1 \times \ldots \times X_k \to \{0,1\}$,*

$$Ee_x[g(x)] \le R(g)^{1/2^k} = \text{(by definition)} \underset{\substack{x_1^0,\ldots,x_k^0,\\ x_1^1,\ldots,x_k^1}}{Ee}\Big[\sum_{\varepsilon_1,\ldots,\varepsilon_k \in 0,1} g(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k})\Big]^{1/2^k}.$$

Proof. Recall the Cauchy-Schwarz inequality: for every random variable X: $E[X^2] \ge E[X]^2$. This holds because $0 \le E[(X - E[X])^2] = E[X^2] - E[X]^2$. Also recall that if X, X' are independent then $E[X \cdot X'] = E[X] \cdot E[X']$. We have:

$$\underset{x_1,\ldots,x_k}{Ee}[g(x_1,\ldots,x_k)]^2 = \underset{x_1,\ldots,x_k}{E}\,[\underset{x_k}{Ee}[g(x_1,\ldots,x_k)]]^2 \le \underset{x_1,\ldots,x_{k-1}}{E}\,[\underset{x_k}{Ee}[g(x_1,\ldots,x_k)]^2]$$

$$= \underset{x_1,\ldots,x_{k-1}}{E}\,[\underset{x_k^0,x_k^1}{Ee}[g(x_1,\ldots,x_{k-1},x_k^0) + g(x_1,\ldots,x_{k-1},x_k^1)]].$$

The lemma follows by repeating this k times. □

Lemma 15. *For every function $f : X_1 \times \ldots \times X_k \to \{0,1\}$, and every protocol* p^*,*

$$R(f \oplus p^*) = R(f),$$

where $f \oplus p^*$ simply is the function whose output is the XOR of f and p^*.

Proof. Suppose $p^*(x) = g_1(x) + \ldots + g_k(x)$, where g_i is a cylinder in the i-th dimension. We show $\forall f, R(f \oplus g_k) = R(f)$; the same reasoning works for the other coordinates. Note for every x,

$$\begin{aligned}
&\sum_{\varepsilon_1,\ldots,\varepsilon_k \in \{0,1\}} (f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}) + g_k(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}))\\
&= \sum_{\varepsilon_1,\ldots,\varepsilon_k} f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}) + \sum_{\varepsilon_1,\ldots,\varepsilon_k} g_k(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k})\\
&= \sum_{\varepsilon_1,\ldots,\varepsilon_k} f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}) + \sum_{\varepsilon_1,\ldots,\varepsilon_k} g_k(x_1^{\varepsilon_1},\ldots,x_k^{0})\\
&= \sum_{\varepsilon_1,\ldots,\varepsilon_k} f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}) + 2\sum_{\varepsilon_1,\ldots,\varepsilon_{k-1}} g_k(x_1^{\varepsilon_1},\ldots,x_k^{0})\\
&= \sum_{\varepsilon_1,\ldots,\varepsilon_k} f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k}) \mod 2,
\end{aligned}$$

where the second equality holds because g_k does not depend on x_k. □

The straightforward combination of the three lemmas in this section proves Theorem 11.

2.4 $R(GIP) \le 2^{-\Omega(n/2^k)}$

Recall for $f : X_1 \times \ldots \times X_k \to \{0,1\}$ we define

$$R(f) := \underset{\substack{x_1^0,\ldots,x_k^0,\\ x_1^1,\ldots,x_k^1}}{Ee}\left[\sum_{\varepsilon_1,\ldots,\varepsilon_k \in \{0,1\}} f(x_1^{\varepsilon_1},\ldots,x_k^{\varepsilon_k})\right],$$

and that the function GIP is defined as $GIP(x_1, \ldots, x_k) := \sum_i \prod_j (x_j)_i \ mod\ 2$, where $x_i \in \{0,1\}^n$. We have:

$$R(GIP) = \underset{\binom{x_1^0,\ldots,x_k^0,}{x_1^1,\ldots,x_k^1}}{E e} \left[\sum_{\varepsilon_1,\ldots,\varepsilon_k \in \{0,1\}} \sum_i \prod_j (x_j^{\varepsilon_j})_i \right] = E \prod_i e \left[\sum_{\varepsilon_1,\ldots,\varepsilon_k} \prod_j (x_j^{\varepsilon_j})_i \right]$$

$$= E e \left[\sum_{\varepsilon_1,\ldots,\varepsilon_k} \prod_j (x_j^{\varepsilon_j})_1 \right]^n = R \left(\bigwedge_k \right)^n,$$

using in the last equality the fact that any two independent random variables X, Y satisfy $E[X \cdot Y] = E[X] \cdot E[Y]$, and where $\bigwedge_k$ is the AND function on k bits.

To save in notation let us replace $(x_1^0)_1, \ldots, (x_k^0)_1$ with $(y_1^0, \ldots, y_k^0)$, where $(y_i^0) \in \{0,1\}$; and similarly for $(x_1^1)_1, \ldots, (x_k^1)_1$. So we have:

$$R(GIP) = \underset{\binom{y_1^0,\ldots,y_k^0,}{y_1^1,\ldots,y_k^1}}{E e} \left[\sum_{\varepsilon_1,\ldots,\varepsilon_k \in \{0,1\}} \prod_j y_j^{\varepsilon_j} \right]^n.$$

Suppose that $y_1^0 \neq y_1^1, \ldots, y_k^0 \neq y_k^1$; then there exists exactly one choice of $\varepsilon_1, \ldots, \varepsilon_k$ making $\prod_j y_j^{\varepsilon_j} = 1$, and consequently

$$e \left(\sum_{\varepsilon_1,\ldots,\varepsilon_k} \prod_j y_j^{\varepsilon_j} \right) = e(1) = -1.$$

We have $y_1^0 \neq y_1^1, \ldots, y_k^0 \neq y_k^1$ with probability 2^{-k}. Therefore:

$$R(GIP) = E e \left[\sum \prod_j y_j^{\varepsilon_j} \right]^n \leq (-1 \cdot 2^{-k} + 1 \cdot (1 - 2^{-k}))^n = (1 - 2^{-k+1})^n \leq e^{-\Omega(n/2^k)}.$$

This concludes our bound on $R(GIP)$ and also the proof of Theorem 10.