CS294: Probabilistically Checkable and Interactive Proofs January 31, 2017

Interactive Proofs with Bounded Communication/Randomness

Instructor: Alessandro Chiesa & Igor Shinkar *Scribe: Peter Manohar*

Suppose we restrict the number of bits sent by the prover to be only 1 bit. One might guess that the set of languages that have a 1 bit IP is simple, say something like $\mathcal{P}$. However, the IP for GNI is a protocol where the prover only sends one bit, so clearly this is not the case. The set of languages which have a 1 bit IP is not trivial. We would like to know the relation between the time complexity of a language and the communication complexity of its IP. We will prove 4 statements about IP's with restricted communication complexity. All four theorems can be found in [GH98].

1 The Theorems

Theorem 1 *Suppose L has an IP system where on input x, the verifier flips at most $c = c(|x|)$ bits and the number of bits sent between verifier and prover is at most c. Then $L \in \mathrm{DTime}(2^{O(c)}\mathrm{poly}(|x|))$.*

Theorem 2 *Suppose L has an IP system where on input x the number of bits exchanged is at most c. Then $L \in \mathrm{BPTime}(2^{O(c)}\mathrm{poly}(|x|))$.*

Theorem 3 *Suppose L has a public coin IP system such that on input x, the number of bits sent from the prover is at most c. Then $L \in \mathrm{BPTime}(2^{O(c \log c)}\mathrm{poly}(|x|))$. We'll only prove that $L \in \mathrm{BPTime}(2^{O(c)}c^{O(r)}\mathrm{poly}(|x|))$, where r is the round complexity.*

Theorem 4 *If the IP is not public coin in theorem 3, then $L \in \mathrm{BPTime}(2^{c \log c}\mathrm{poly}(|x|))^{\mathrm{NP}}$.*

The Main Idea

We will make a tree for the proof system. For a fixed input x, the tree T_x is the tree where each path in the tree is a transcript of the protocol. . The root of the tree is the root, and each edge from the root are the possible messages that the verifier can send. All edges from level 1 to 2 are the possible responses of the prover to the message sent by the verifier, etc. Each of the verifier's edges are have a label. the label is the message sent and the probability of that message being sent. Each prover's edge is just the message sent by the prover.

Definition 5 *The value of the tree $\mathrm{val}(T_x)$ is the max probability of the verifier accepts, i.e. $\max_P \Pr[V \mathrm{accepts}]$. The max is taken over all possible P.*

We can compute $\mathrm{val}(T_x)$ recursively by assigning a value to each vertex in the tree. The value of the root will be $\mathrm{val}(T_x)$. For each leaf, the value of the leaf is 0 if the verifier rejects on that transcript and 1 if the verifier accepts. For vertices where the outgoing edges are prover edges, the value is $\max_{\mathrm{children}\ c} \mathrm{val}(c)$. For vertices where the outgoing edges are verifier edges, $\mathrm{val} = \mathbb{E}_{\mathrm{children}\ c}[\mathrm{val}(c)]$.

Now we can prove the four theorems.

Proof of Theorem 1: For every input x, the size of the tree, $|T_x|$, is $2^{O(c)}$. Therefore, we can construct T_x and compute the value $\mathrm{val}(T_x)$ in $2^{O(c)}\mathrm{poly}(|x|)$ time. Since we know the $\mathrm{val}(T_x)$, it is easy to distinguish between the cases when $x \in L$ and $x \notin L$, as in the first case $\Pr[V \text{accepts}] \geq \frac{2}{3}$ and in the second case $\Pr[V \text{accepts}] \leq \frac{1}{3}$. □

Proof of Theorem 2: Now, we can no longer construct the tree T_x, as it might be too large. Instead, we approximate the tree. We sample $m = 2^{O(c)}$ random strings for the randomness of the verifier, and look at the subtree A_x where we only consider verifier edges that correspond to these random strings. Note that $|A_x| \leq m2^c$, so we can compute $\mathrm{val}(A_x)$ in time $m2^c\mathrm{poly}(|x|)$. We prove the following claim, from which the theorem follows.

Claim 6 *With high probability* $|\mathrm{val}(A_x) - \mathrm{val}(T_x)| < 0.01$.

Proof: Let's prove that for all provers, $|\mathrm{val}_P(A_x) - \mathrm{val}_P(T_x)| < 0.01$ with probability $1 - 2^{-\Omega(m)}$. Fix a prover P, and let $X_1, \ldots, X_m$ be random variables where X_i is 1 if the verifier accepts while interacting with P when the random string is the ith random string that we sampled. Note that the "randomness" of these variables comes from the sampling of the random strings. These are independent random variables, and each has mean $\mathrm{val}_P(T_x)$. By applying a Chernoff bound, we get that, $\Pr[|\mathrm{val}_P(A_x) - \mathrm{val}_P(T_x)| > 0.01] < 2^{-2\cdot(0.1)^2 m}$. There are at most $(2^c)^{2^c}$ provers, as for any prefix of the protocol (which is all of the bits sent between verifier and prover) there are 2^c combinations, and at most c more bits are sent. Therefore, the probability that there exists a prover such that $|\mathrm{val}_P(A_x) - \mathrm{val}_P(T_x)| > 0.01$ is at most (by taking a union bound over all provers)

$$2^{-2\cdot(0.1)^2 m} \cdot (2^c)^{2^c} = 2^{-0.02m + c2^c} < 0.01$$

since $m = 2^{O(c)}$. □

Since our estimate for $\mathrm{val}(T_x)$ is within 0.01 of the true value, we can easily distinguish between the case when $\mathrm{val}(T_x) \geq 0.67$ and $\mathrm{val}(T_x) \leq 0.33$, so we are done. □

Proof of Theorem 3: The only restriction is now that number of bits sent by the prover, and the IP must be public coin. Therefore, our bound on the number of provers no longer works and so the previous proof fails when we take the union bound over all provers. We will again approximate T_x by a subtree A_x. For each vertex in T_x that is a verifier vertex, we'll take $s = O(c^4)$ random children (with replacement). For the prover vertices, we take all of the children. Now, $|A_x| \leq 2^c s^r = 2^c c^{O(r)}$, so we can compute $\mathrm{val}(A_x)$ in $2^c c^{O(r)}\mathrm{poly}(|x|)$ time. So, we just need to show that $\mathrm{val}(A_x) \approx \mathrm{val}(T_x)$.

Claim 7 $\Pr[|\mathrm{val}(A_x) - \mathrm{val}(T_x)| < 0.01] > 0.99$

Proof: The main idea is to use hybrid trees. Define $H_0 = T_x$, $H_{r+1} = A_x$, and H_i is the tree where we take the first $2i + 1$ layers from A_x and the rest from T_x. For every i, we claim that $\Pr[|\mathrm{val}(H_i) - \mathrm{val}(H_{i+1})| > \frac{0.1}{r+1}] < \frac{0.01}{r+1}$. The number of vertices in A_x is at most $2^c s^r$, as there are at most 2^c prover vertices and s^r verifier vertices. The claim follows by showing that (as in Theorem 2) the difference between $\mathrm{val}(H_i)$ and $\mathrm{val}(H_{i+1})$ is the same as the difference between the expected value of a random variable and the mean of s samples of the random variable, and then taking a Chernoff bound to conclude. Details are in [GH98]. □ □

Remark 8 *Theorem 3 impiles that we cannot expect IPs for NP languages where the prover sends only a few bits, at least in the public coin setting.*

Proof of Theorem 4: Now, the IP is not public coin. The problem is that we can no longer construct A_x, because we cannot sample the verifier's next message just from the transcript of the protocol as it may depend on some private coins already flipped by the verifier, but not present in the transcript. This problem is solved by the NP oracle. The NP oracle is used to sample a random pad for the verifier that is consistent with the current transcript of the protocol. Using this random pad, we can then sample the verifier's next message, and continue as in Theorem 3. □

2 Perfect Completeness for AM Protocols

We have already seen that GNI is in $AM_{(\frac{2}{3},\frac{1}{3})}$. We wish to achieve perfect completeness.

Theorem 9 $AM_{(\frac{2}{3},\frac{1}{3})} \subset AM_{(1,\frac{1}{2})}$

Proof: There are 3 steps to prove.

1. $AM_{(\frac{2}{3},\frac{1}{3})} \subset MAM_{(1,\frac{1}{4})}$
2. $MA_{(1,\frac{1}{2})} \subset AM_{(1,\frac{1}{2})}$
3. Use 2 to show that $MAM \subset AMM = AM$.

We will only show step 1, i.e. amplification. Suppose we have an AM protocol that tosses R coins. We repeat the protocol t times in parallel to get an AM protocol with completeness $1-\exp(-t)$ and soundness $\exp(-t)$. We need to figure out a way to use the first Merlin to get perfect completeness. If $x \in L$, we want the verifier to always be convinced. So far, the set of random strings where the verifier convinced is $1-\exp(-t)$ of all strings. Let $r = tR$. We use the first Merlin in the following protocol.

Protocol: MAM

- Prover samples $y_1, \ldots, y_r \in \{0,1\}^r$ and sends them to the verifier.
- Verifier samples $c \in \{0,1\}^r$ and sends c to the prover.
- The prover sends back a proof $\pi_{c \oplus y_i}$ for each $i \in \{1, \ldots, r\}$, and the verifier accepts if at least one of the proofs passes.

Lemma 10 *$\exists y_1, \ldots, y_r$ such that $\forall c$, $\exists i$ such that verifier will accept the proof $\pi_{c\oplus y_i}$.*

□

References

[GH98] Oded Goldreich and Johan Håstad, *On the complexity of interactive proofs with bounded communication*, Information Processing Letters **67** (1998), no. 4, 205–214.