Undergraduate Complexity Theory
Lecture 25: Interactive Proofs: $\mathsf{IP} = \mathsf{PSPACE}$

Marcythm

July 29, 2022

1 Lecture Notes

past lectures: complexity theory from its inception in the mid 60s up to the end of the 80s (space, time, randomness). next: get close to future, a glimpse past the 80s.

1. today: I.P. (early 90s)
2. Thurs: average case hardness, hardness of approximation, Feige's Hypothesis (early '00s)
3. next Tues: Hardness within P (mid 2010's)
4. next Thurs: why P vs NP hard? (1975)
   "people are frustrated by the inability to prove it, they started proving theorems about why it's hard to prove it b/c they had nothing else they could do."

Proof Systems: statement S (e.g. $x \in L$), all-powered (i.e. can do computation in arbitrary time) prover P who always try to convince verifier that S is true by giving proof y, verifier V who will do some poly-time computation on the statement S and the proof y given by P where

$$\begin{aligned} x \in L &\implies \exists y : V(x, y) = 1 \\ x \notin L &\implies \forall y : V(x, y) = 0 \end{aligned}$$

Those language L with such a proof system are those in NP.

Interactive Proof System: the prover and the verifier are allowed to interact like: P send a proof y_1 to V, V send message q_1 back to P which is thought to be a question, P answers the question with y_2, ..., and after poly # of rounds V do the poly-time computation $V(y_1, q_1, \ldots, y_n, q_n)$ which satisfies similar requirements. (all messages are in poly length.)

Fact 1.1. *If the verifier is deterministic, there are no difference between interactive proof system and the non-interactive one, since the prover can predict all the question V will ask, and prepare the answers, send them all in the first round.*

What if V is **randomized**? Consider $\overline{\mathsf{GISO}}$, where $\mathsf{GISO} := \mathsf{GRAPH\text{-}ISOMORPHISM}$. $\overline{\mathsf{GISO}}$ has an interactive proof system: V sends to P a randomly relabeled version H of a randomly picked (from the two) graph G_i, and ask P to guess which graph this is. If P guessed correctly, then accept, otherwise reject. For the case that input $\langle G_0, G_1 \rangle \in \overline{\mathsf{GISO}}$, $\mathbf{Pr}[V \text{ accepts}] = 1$. If $\langle G_0, G_1 \rangle \notin \overline{\mathsf{GISO}}$, $\mathbf{Pr}[V \text{ accepts}] = 1/2$ whatever the strategy P choose.

Definition 1.2. $\mathsf{IP}[k]$ contains those language L having a k-round i.p. system where

$$\begin{aligned} x \in L &\implies \exists \text{ Prover strategy s.t. } \mathbf{Pr}[V(\ldots) = 1] \geq 2/3 \\ x \notin L &\implies \forall \text{ Prover strategy s.t. } \mathbf{Pr}[V(\ldots) = 1] \leq 1/3 \end{aligned}$$

Definition 1.3. $\mathsf{IP} = \mathsf{IP}[\text{poly}(n)]$. The class is intuitively those having efficient interactive proof systems.

Observation 1.4. $\mathsf{NP} \subseteq \mathsf{IP}[1]$.

Observation 1.5. $\mathsf{BPP} \subseteq \mathsf{IP}[0]$.

Fact 1.6. *Randomized provers doesn't "help", i.e. won't have ability to prove any new language.*

Fact 1.7. *For fixed verifier, given some x, can compute $\max_{\text{prover strategy}} \mathbf{Pr}[V \leftarrow P \text{ acc on } x]$ w/ $\text{poly}(|x|)$ space.*

Corollary 1.8. $\mathsf{IP} \subseteq \mathsf{PSPACE}$. *i.e. prover can be modeled by a poly-space TM.*

Theorem 1.9. *If change the $2/3$ in the definition of IP to 1, it doesn't change IP. i.e. you can automatically upgrade two-sided error i.p. system to one-sided error one.*

Fact 1.10. *If the verifier shows the result of its random coin flips to prover, it doesn't change IP too.*

In '80s, people thought IP is "not much more" than "randomized NP". In '88, Fortnow-Sipser conjectured $\overline{\mathsf{3SAT}} \notin \mathsf{IP}$. But in '89, LFKN proved $\overline{\mathsf{3SAT}} \in \mathsf{IP}$ (a stronger result: $\#\mathsf{3SAT} \in' \mathsf{IP}$). Two weeks later, Shamir '89 showed that $\mathsf{TQBF} \in \mathsf{IP}$, i.e. $\mathsf{PSPACE} \subseteq \mathsf{IP}$, hence $\mathsf{PSPACE} = \mathsf{IP}$.

Proof sktech of $\#\mathsf{3SAT} \in' \mathsf{IP}$: for formula ϕ with n variables and m clauses, construct polynomial $q(x_1, \ldots, x_n)$, e.g. $(1-(1-x_1)(1-x_3)x_5)(1-x_2x_4(1-x_6))\cdots$ for $(x_1 \vee x_3 \vee \overline{x_5}) \wedge (\overline{x_2} \vee \overline{x_4} \vee x_6) \wedge \cdots$. Thus we can convert the problem into a more algebraic looking:

$$\sum_{x_1=0}^{1}\sum_{x_2=0}^{1}\cdots\sum_{x_n=0}^{1} q(x_1, x_2, \ldots, x_n) = K?$$

then, prover says here's prime number p of $2n$ bits, and verifier acknowledges. (in '89s only known that $\mathsf{PRIMES} \in \mathsf{NP}$, so a proof should also be given by prover; but now we have $\mathsf{PRIME} \in \mathsf{P}$.)

Fact 1.11. *the equation is true iff it's true under* $\bmod p$.

key brilliant move: consider another univariate polynomial $r(X_1)$ where X_1 is "indeterminate":

$$r(X_1) = \sum_{x_2=0}^{1}\cdots\sum_{x_n=0}^{1} q(X_1, x_2, \ldots, x_n) \bmod p$$

we know $\deg r \le \deg q \le 3m$, and r's coeffs are in $[0, p)$, so the whole r can be written down in $O(mn)$ bits. The verifier then checks $r(0) + r(1) = K$. We're almost done here, only to check that the r verifier received is correct. Here verifier picks a random $a \in [p-1]$, sends it to prover, and ask it to prove that

$$r(a) = \sum_{x_2=0}^{1}\cdots\sum_{x_n=0}^{1} q(a, x_2, \ldots, x_n),$$

which can be done inductively.

Here, if the prover is lying, it must give a wrong r' to verifier, which can only agree with the true r on at most $3m$ values since $\deg(r - r') \le \max(\deg r, \deg r') \le 3m$, so the probability that the verifier is cheated in each step is extremely small since p is about 2^{2n}, and $3m \le 3 \times 2^n << 2^{2n}$.

2 Reading

2.1 sipser 10.4 (Interactive Proof Systems)

Interactive proof systems provide a way to define a probabilistic analog of the class NP, much like probabilistic polynomial time algorithms provide a probabilistic analog to P.

Formal definition of i.p.s.: Verifier $V : \Sigma^* \times \Sigma^* \times^* \to \Sigma * \cup \{accept, reject\}$, where the inputs are: input string (the statement), random input, and partial message history. Similarly, the prover $P : \Sigma^* \times \Sigma^* \to \Sigma^*$.

proof of $\mathsf{IP} = \mathsf{PSPACE}$.