CS 6810: Theory of Computing	**Fall 2021**

Lecture 18: Oct 28, 2021

Lecturer: Eshan Chattopadhyay *Scribe: Conan Gillis*

More on Interactive Proofs

In today's class, we will prove the following theorem that is due to Lund, Fortnow, Karloff and Nisan [LFKN92]. Building on these ideas, Shamir [Sha92] proved that IP=PSAPCE.

Theorem 0.1. $\mathsf{CoNP} \subseteq \mathsf{IP}$

Proof. Recall that $\overline{\mathsf{SAT}}$ is complete for CoNP, so if $\overline{\mathsf{SAT}} \in \mathsf{IP}$, then since IP is closed under polynomial reduction all of CoNP must be in NP. Thus, we aim to show $\overline{\mathsf{SAT}}$ is indeed in IP. Recall $\overline{\mathsf{SAT}} = \{\phi : \phi$ is a 3CNF that is satisfiable$\}$. To this, define $\#\mathsf{SAT} = \{(\phi, \mathrm{r}) : \phi$ is a 3CNF with exactly r satisfying truth assignments$\}$. Observe that $\overline{\mathsf{SAT}} \leq_p \#\mathsf{SAT}$ (take $r = 0$ in the obvious reduction). Thus, $\#\mathsf{SAT} \in \mathsf{IP}$ implies our desired claim of $\overline{\mathsf{SAT}} \in \mathsf{IP}$.

For notation, let ϕ_0 denote ϕ with the first variable x_1 set to 0, and let ϕ_1 denote ϕ with x_1 set to 1. Furthermore, let ϕ_{00} denote ϕ with x_1, x_2 set to 0, and let ϕ_{01} denote ϕ with x_1 set to 0 and x_2 set to 1, with $\phi_{10}, \phi_{11}, \phi_{000}, ...,$ defined analogously. Moreover, in this notation, let $r_{b_1 b_2 b_3 \ldots b_k}$ denote the number of satisfying assignments for $\phi_{b_1 b_2 \ldots b_k}$ ($b_i \in \{0,1\}$), so r_0 is the number of satisfying assignments for ϕ_0, r_1 the number for ϕ_1, and so on. Then, if r is the number of satisfying assignments for ϕ itself, we must have $r = r_0 + r_1$. In particular, this means $(\phi, r) \in \#\mathsf{SAT}$ if and only if $r_0 + r_1 = r$.

We now propose an insufficient, but suggestive, interactive-proof protocol for $\#\mathsf{SAT}$ using a prover P and verifier V. We are given (ϕ, r). First, the prover sets V a pair of numbers, which we call (r_0, r_1) (these may not actually be the real numbers of satisfying assignments for ϕ_0, ϕ_1; this will be important in our analysis of soundness). The verifier first checks $r_0 + r_1 = r$, rejecting if this does not hold, and sends the prover a random bit $b \in \{0,1\}$ if it does. The procedure goes on recursively on the input (ϕ_b, r_b).

For completeness, if $(\phi, r) \in \#\mathsf{SAT}$ then an honest prover it will send (r_0, r_1) that do indeed correspond to ϕ_0, ϕ_1 each time, so the verifier will never reject and thus will (correctly) assert $(\phi, r) \in \#\mathsf{SAT}$.

For soundness (here is where the protocol is insufficient), suppose $(\phi, r) \notin \#\mathsf{SAT}$. Then, at least one of $(\phi_0, r_0), (\phi_1, \phi_1)$ is not in $\#\mathsf{SAT}$. Note that r_0, r_1 are the values returned by the possibly dishonest prover, potentially distinct from the numbers of satisfying assignments for these two 3CNFs. The probability that $(\phi_b, r_b) \notin \#\mathsf{SAT}$ is at least $\frac{1}{2}$, where b is the bit randomly chosen in the protocol. Now, in the worst case scenario, a prover may send "good" values for (r_0, r_1) for all the rounds until the end (if there are "good" values for all rounds, then $(\phi, r) \in \#\mathsf{SAT}$), and there may be only one location (ϕ_i, r_i) at the end of the recursion tree that is not in $\#\mathsf{SAT}$. In this case, the best bound we can make is $P(V$ does not accept $(\phi, r)) \geq \frac{1}{2^n} \iff P(V$ accepts $(\phi, r)) \leq 1 - \frac{1}{2^n}$, which is much too high for our purposes, since we need the latter probability to be bounded above by $\frac{1}{3}$.

To salvage this protocol, we introduce the technique of Arithmetization:

Arithmetization Pick a large prime $p \in (2^n, 2^{n+1}]$. Given a formula ϕ on n variables, we will find a polynomial f_ϕ on n variables in $\mathbb{F}_p$ such that for all $x \in \{0,1\}^n$, $\phi(x) = f_\phi(x) \mod p$. It is, in fact, easy to do so. If ϕ has n variables and m clauses, it is of the form

$$\phi = \Lambda_{i=1}^m C_i$$

where $c_i = l_{a_i} \vee l_{b_i} \vee l_{c_i}$ for $l_{a_i}, l_{b_i}, l_{c_i} \in \{x_1, ..., x_n, \overline{x_1}, ..., \overline{x_n}\}$. If $l_{a_i} = x_j$ for some j, define $g_{a_i} = (1 - X_j)$, otherwise $l_{a_i} = \overline{x_j}$ and we define $g_{a_i} = X_j$ (here, x_j is a literal in ϕ, X_j is some variable ranging over $\mathbb{F}_P$). Define g_{b_i}, g_{c_i} analogously to g_{a_i}, and let $f_i = 1 - g_{a_i} g_{b_i} g_{c_i}$. Then,

$$f_\phi = \Pi_{i=1}^m f_i.$$

As an example of this, if

$$C_i = x_2 \vee \overline{x_5} \vee x_{20},$$

then

$$f_i = 1 - (1 - X_2) X_5 (1 - X_{20}).$$

Observation 0.2.

$$r = \sum_{x \in \{0,1\}^n} f_\phi(x) \mod p$$

$$deg(f_\phi) \leq 3m.$$

We now solve a new problem $\#\mathsf{POLY} = \{(\mathsf{f}, \mathsf{r})$: f a polynomial in m variables over $\mathbb{F}_p$ of degree d at most polynomial in n, for $p \in (2^n, 2^{n+1}]$,

$$r = \sum_{x \in \{0,1\}^n} f(x) \mod p\}.$$

By the above arithmetization argument, $\#\mathsf{SAT} \leq_\mathsf{p} \#\mathsf{POLY}$, so showing $\#\mathsf{POLY} \in \mathsf{IP}$ will be enough to complete our proof.

Our interactive proof protocol (inspired by the one described above) is as follows. Given (f, r), the prover first sends the verifier a polynomial

$$g(x_1) = \sum_{\substack{x_2 \in \{0,1\} \\ \vdots \\ x_n \in \{0,1\}}} f(x_1, x_2, ..., x_n).$$

In other words, g is the sum of the values of f as a polynomial in x_1 for all possible choices of $x_2, ..., x_n$ (at least, that is what the verifier wants it to be - the prover could still be dishonest). The verifier then checks if $g(0) + g(1) = r$, rejecting if not, then chooses a random $\lambda_1 \in \mathbb{F}_p$ and recursing on $(f(\lambda_1, x_2, ..., x_n), g(\lambda_1))$.

Claim 0.3. *This works*

Proof. Suppose $(f, r) \in \#\mathsf{POLY}$. If the prover is honest, the protocol will not reject (f, r), so completeness holds.

Otherwise, suppose

$$r \neq \sum_{x \in \{0,1\}^n} f(x) \mod p$$

and $h(x_1) = \sum_{\substack{x_2 \in \{0,1\} \\ \vdots \\ x_n \in \{0,1\}}} f(x_1, x_2, ..., x_n)$. We see $h(0) + h(1) = \sum_{x \in \{0,1\}^n} f(x)$ and $g(0) + g(1) = r$, so $g \neq h$. Therefore, $P_{\lambda_1}[g(\lambda_1) = h(\lambda_1)] = P_{\lambda_1}[(g - h)(\lambda_1)] = 0 < \frac{d}{p}$. Therefore, $P[$ verifier rejects$] \geq (1 - \frac{d}{p})^n \geq 1 - \frac{dn}{p}$ so $P_{\lambda_1}[$ prover accepts $(f, r)] \leq \frac{dn}{p} < \frac{1}{3}$ since $p > 2^n$ and dn is poly(n). This completes the proof. □

□

References

[LFKN92] Carsten Lund, Lance Fortnow, Howard Karloff, and Noam Nisan. Algebraic methods for interactive proof systems. *Journal of the ACM (JACM)*, 39(4):859–868, 1992.

[Sha92] Adi Shamir. Ip= pspace. *Journal of the ACM (JACM)*, 39(4):869–877, 1992.