COM S 6810 Theory of Computing January 10/12, 2009

Lecture 15,16: IP=PSPACE

Instructor: Rafael Pass *Scribe: Ashwinkumar B. V*

1 IP=PSPACE

Theorem 1 *[1, 2] IP=PSPACE*

Proof. The proof of the theorem is by proving $IP \subseteq PSPACE$ and $PSPACE \subseteq IP$.

1. Since the prover can use an arbitrary function, it can in principle use unbounded computational power (or even compute undecidable functions). It is easy to see that given any verifier V , we can compute the optimum prover (which, given x, maximizes the verifiers acceptance probability) using $poly(|x|)$ space (and hence $2^{poly(|x|)}$ time).This is by enumerating each possible communication pattern and computing the probability of acceptance of verifier. Thus $IP \subseteq PSPACE$.

2. As a motivating example we first prove $\#P \subseteq IP$ by giving an interactive protocol for $\#3SAT$. i.e Given $\{\phi, k'\}$ an interactive protocol is given to prove $\#\phi \geq k'$. Here first the prover sends $k \geq k'$ and then proves that $\#\phi = k$. For this we use Arithmetization defined shortly.

   Definition 1 *Arithmetization:- Give a boolean formula $\phi(\vec{x})$ to give a polynomial $g(\vec{x})$ such that $\forall \vec{x} \in \{0,1\}^n, \phi(\vec{x}) = g(\vec{x})$.*

   First rewrite ϕ with only $\wedge$(and) and $\neg$(not) and remove all $\vee$(or). Once this is done ϕ is defined recursively. If x,y are clauses and X,Y are corresponding polynomials then polynomial corresponding to $x \wedge y$ is XY and polynomial corresponding to $\neg x$ is $1 - X$. It is easy to see that these polynomials can be constructed in polynomial time and that they satisfy the requirement for Arithmetization.

   We can easily see that given ϕ we have that $\#\phi = \sum_{\vec{x} \in \{0,1\}^n} g(\vec{x}) mod(p)$ where $p > 2^n$ and p is prime. Hence the protocol is complete if we give an IP which proves $k = \sum_{x_1 \in \{0,1\}} \ldots \sum_{x_n \in \{0,1\}} g(\vec{x}) mod(p)$. We do this by giving an IP recursively.

   - P computes a prime p such that $2^n < p \leq 2^{2n}$ and sends it to V. V checks if p is prime and rejects if it is not.
   - P proves to V using subset sum protocol that $\sum_{x_1 \in 0,1} \ldots \sum_{x_n \in 0,1} g(\vec{x}) mod(p) = k$

Definition 2 *Subset sum protocol - IP to prove that* $\sum_{x_1 \in 0,1} \dots \sum_{x_n \in 0,1} g(\vec{x}) mod(p) = k$

- Base case:- If the equation has only one variable, V checks the equality manually else it follows the below steps.
- P sends $S(x_1) = \sum_{x_2 \in 0,1} \dots \sum_{x_n \in 0,1} g(\vec{x}) mod(p)$. V checks $S(0) + S(1) = k$ and rejects if it is not.
- V selects a random number $a \in \{0, 1, \dots, p-1\}$ and sends it to P.
- P recursively proves V that $S(a) = \sum_{x_2 \in 0,1} \dots \sum_{x_n \in 0,1} g(a, \vec{x}_{-1}) mod(p)$

Note that soundness and completeness of the IP for $\#SAT$ is the same as that of the IP for subset sum. We now prove the soundness and completeness.

- Communication and computation by verifier is polynomial - Trivally seen.
- Completeness - Trivially satisfied with probability 1 as, if the Prover acts according to the protocol, each of the steps must be trivially satisfied.
- Soundness - Satisfied with probability atleast $1 - \frac{3mn}{p}$. The proof is by induction. The base case trivially holds as V checks any equation with one variable explicitly and hence the equation proved must be true with probability 1. Now for the induction step assume that equation proved recursively with n-1 variables holds with probability atleast $1 - \frac{3m(n-1)}{p}$. Then the equation with n variables can fail due to two reasons. One is that the recursion step fails. This happends with probability at most $\frac{3m(n-1)}{p}$.
  The other being the fact that the equation chosen by prover matches with the correct equation at the point a. This can only happen if the polynomial got by subtracting the correct equation with that of equation chosen by prover has a as its root. As we know that the degree of polynomial is at most 3m and that a polynomial with degree d has atmost d roots we can conclude that this does happen with probability atmost $\frac{3m}{p}$. Hence by union bound the protocol fails with probability atmost $\frac{3mn}{p}$.

3. To prove that $PSPACE \in IP$ it is enough to give a IP for TBQF(True boolean quantified formula) as we already know that TBQF is a complete problem for PSPACE. We give two different proofs for this. The first of these is by the Turing award winner Shamir.

   We first note that any TBQF can be written as $\forall x_1, \exists x_2, \dots, Q_n x_n, \phi(\vec{x})$ holds. (where Q is either $\forall$ or $\exists$). Now it is easy to see that by applying arithmetization this is equivalent to writing the formula as $\prod_{x_1} \sum_{x_2} \dots \mathbb{Q}_{x_n} P_\phi(\vec{x}) > 0?$. But it is easy to see that this equation cannot be proved due to many reason each of which will be overcome in steps.

- It is easy to see that the equation can have value which is $O^\star(2^{2^n})$. This can be over come by asking the prover to prove the equation modulus a prime number $1 < p \leq 2^{poly(n)}$. Such a number should exist because if every prime $< 2^{poly(n)}$ divides the value then it is easy to see that the value will be greater than $O^\star(2^{2^n})$.
- We modify the boolean formula such that the degree of the polynomial is not too large at each step as follows. The boolean formula is re-written from right to left by considering each quantifier one at a time and adding many more quantifiers and variables in the process. Note that only the quantifiers present in the original formula are considered and not the newly added ones.

  If the quantifier being considered in $\forall$ then do nothing. Else represent the formula being considered as $\phi(x_1, x_2, \ldots, x_{i-1}) = \forall x_i \phi(x_1, x_2, \ldots, x_i)$. Now rewrite the formula as $\phi'(x_1, x_2, \ldots, x_{i-1}) = \forall x_i \exists x_1^i, x_2^i, \ldots, x_n^i \bigwedge_{j=1}^{n} x_j = x_j^i \wedge \phi(x_1^i, x_2^i, \ldots, x_i^i)$. It is easy to see that both the boolean formulae's are equivalent to each other. Now it is easy to see that if we arithmetize this formula each variable can have degree atmost $3m$ and hence the subset sum protocol can be used by the prover to prove the arithmetized formula.
- Another proof for $TBQF \in IP$. Here we do not change the given boolean formula, but change the Arithmetization so that the degree of the resulting polynomial is not too huge. i.e Given the TBQF our new arithmetized polynomial will be

$$\prod_{x_1} R_{x_1} \sum_{x_2} R_{x_1} R_{x_2} \prod_{x_3} \ldots \mathbb{Q}'_{x_{n-1}} R_{x_1} R_{x_2} \ldots R_{x_{n-1}} \mathbb{Q}_{x_n} P_\phi(\vec{x}) > 0? \tag{1}$$

  Here R_{x_i} is a degree reduction operator which takes a polynomial of arbitrary degree and converts it to a polynomial of degree 1 such that both match at points 0 and 1. i.e $R_{x_i}\{P(x_i)\} = (1 - x_i).P(0) + x_i.P(1)$. Now it is trivial fact to check that all the properties will be satisfied if we run the subset sum protocol.

2 AM[k]=AM

We recollect the definition of some of the interactive protocols.

- AM[k]=the class of languages that can be decided by a k round interactive proof in which each verifiers message consists of sending a random string of polynomial length, and these messages comprise of all the coins tossed by the verifier. A proof of this form is called a public coin proof.
- AM=AM[2]

- MA=the class of languages with 2-round public coins interactive proof with the prover sending the first message. That is, $L \in MA$ if there is a proof system for L that consists of the prover first sending a message, and then the verifier tossing coins and applying a polynomial-time predicate to the input, the proverś message and the coins.

We prove that $MA \subseteq AM$.Using similar techniques one can extend the proof that for any constant k we have that $AM[k] = AM$. Let $L \in MA$, to prove that $L \in AM$. Let $\mathcal{E}_{MA}$ and $\mathcal{E}_{AM}$ be corresponding executions.

Consider the protocol $\mathcal{E}_{MA}$ for L with completeness 1 and soundness $2^{-2|m|}$(this can be acheived by parallel repetition){Note that $|m|$ does not increase in $\mathcal{E}_{MA}$ due to repetition}. Let m be message sent by P to V in round 1 and r be random coin tosses. Then by completeness $x \in L$ if $\exists P$ such that $V(m_P, x) = 1$ with probability 1 and by soundness $x \notin X$ then $\forall P$ we have that $V(m_P, x) = 0$ with probability $2^{-2|m|}$. Define $\mathcal{E}_{AM}$ such that in round one, V sends r to P and in round two P sends m_P to V.

It is quite trivial to see that completeness holds in $\mathcal{E}_{AM}$. Note that for each value of m_P soundness condition holds with probability $2^{-2|m|}$(as it holds in $\mathcal{E}_{AM}$). Hence by union bound the soundness condition holds for $\mathcal{E}_{AM}$ with probability $\frac{2^{|m|}}{2^{2|m|}} \leq 2^{-|m|}$. (Note that m_P is a specific instance of message m)

References

[1] Carsten Lund, Lance Fortnow, Howard J. Karloff, Noam Nisan. Algebraic Methods for Interactive Proof Systems. J. ACM 39(4): 859-868 (1992)

[2] Adi Shamir. IP = PSPACE. J. ACM 39(4): 869-877 (1992)