IITM-CS6845: Theory Toolkit February 16, 2012

Lecture 22-23 : PSPACE $\subseteq$ IP

Lecturer: Jayalal Sarma.M.N. *Scribe: Sivaramakrishnan.N.R.*

THEME: Between P and PSPACE

1 Interactive Protocol for #SAT

In the previous class we looked at Interactive Protocol for Permanent of a Martix. With that we concluded that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{IP}$. In this section we will present an Interactive Protocol for #SAT which can be extended to prove $\mathsf{PSPACE} \subseteq \mathsf{IP}$.

1.1 Arithmetization of Boolean Formula

Given a boolean formula ϕ, we produce an formula $\widetilde{\phi}$ such that

$$\phi(x_1, x_2, ..., x_k) \; is \; satisfiable \Leftrightarrow \widetilde{\phi}(x_1, x_2, ..., x_k) = 1.$$

The expression ϕ is said to have been arithmetized .
The Procedure:
Basic Building blocks:

- $\widetilde{x \wedge y} \rightarrow xy$.
- $\widetilde{\neg x} \rightarrow 1 - x$.
- $\widetilde{x \vee y} = x + y - xy$ (De Morgan's law).

Recursive Procedure:

- $\widetilde{\phi_1 \wedge \phi_2} \rightarrow \widetilde{\phi_1}\widetilde{\phi_2}$.
- $\widetilde{\neg\phi} \rightarrow 1 - \widetilde{\phi}$.
- $\widetilde{\phi_1 \vee \phi_2} \rightarrow \widetilde{\phi_1} + \widetilde{\phi_2} - \widetilde{\phi_1}\widetilde{\phi_2}$.

Claim 1. *The above algorithm yields ,*$\phi(x_1, x_2, ..., x_k)$ *is satisfiable* $\Leftrightarrow \widetilde{\phi}(x_1, x_2, ..., x_k) = 1$.

Claim 2. $\#\phi = \sum_{x_1 \in \{0,1\}} \sum_{x_2 \in \{0,1\}} .... \sum_{x_k \in \{0,1\}} \widetilde{\phi}(x_1, x_2, ..., x_k)$, *where k is the number of variables in the given boolean expression.*

Proof. The proof of the above two claims is left as an exercise for the readers to verify. □

The above arithmetization has been performed so that the expression obtained can be viewed as a polynomial in variables which helps us in designing an Interactive protocol.

1.2 The protocol

Let n be the size of the formula and k be the number of boolean variables. We observe that the degree d of any variable x_i in $\widetilde{\phi}$ is such that $d \leq n$ as x_i is multiplied atmost n times in the transformation from ϕ to $\widetilde{\phi}$.
Input ϕ.
$'Prover'$: Commits the value of $\#\phi = q$.
Now the verifier has to check if the 'Prover' has indeed given the correct value of q. For doing that the 'Verifier' asks for the polynomial

$$f(x) = \widetilde{\phi}(x_1) = \sum_{x_2 \in \{0,1\}} .... \sum_{x_k \in \{0,1\}} \widetilde{\phi}(x_1, x_2, ..., x_k).$$

The degree d being atmost n, $f(x)$ has atmost $n+1$ coefficients and the 'Prover' can send the coefficients. Now the 'Verifier' can compute $f(0) + f(1)$ and check if it equals q. If it does not equal q, then the 'Prover' has cheated and hence the 'Verifier' rejects. If the sum equals q, then the 'Prover' could have been right or could have cheated by sending a polynomial $f_1(x)$. Now the task of the 'Verifier' is to check if the polynomial $f_1(x)$ is indeed correct. For this the 'Verifier' chooses $r_1 \in_R S \subseteq \mathbb{F}$. Now the 'Verifier' asks the 'Prover' to commit the polynomial

$$f_2(x_2) = \sum_{x_3 \in \{0,1\}} .... \sum_{x_k \in \{0,1\}} \widetilde{\phi}(r_1, x_2, ..., x_k).$$

$$The\ Pr[Error\ at\ stage\ 1] \leq \frac{n}{|S|}\ (By\ Schwartz - Zipple\ Lemma).$$

At p^{th} stage, the 'Verifier' chooses $r_p \in_R S \subseteq \mathbb{F}$ and asks the 'Prover' to commit on the polynomial

$$f_{p+1}(x_{p+1}) = \sum_{x_{p+1} \in \{0,1\}} .... \sum_{x_k \in \{0,1\}} \widetilde{\phi}(r_1, r_2, .., r_p, x_{p+1}..., x_k).$$

Therefore

$$Pr[An\ Error\ has\ occured] = Pr[Error\ has\ occured\ at\ atleast\ one\ stage]$$
$$\leq \frac{nk}{|S|}(Union\ Bound).$$

The size of S can be chosen appropriately to bound the error.

2 PH $\subseteq$ IP

In the previous section we saw that $\mathsf{P}^{\#\mathsf{P}} \subseteq \mathsf{IP}$. That implies that $\mathsf{PH} \subseteq \mathsf{IP}$. But we try to see if the protocol used for #SAT works for PH too.

2.1 Arithmetizing Quantifier Expression

Say

$$\Psi = \exists_{y_1} \forall_{y_2} \forall_{y_3} .... \exists_{y_k} \phi(y_1, ..., y_k),$$

then

$$\#\Psi = \sum_{y_1 \in \{0,1\}} \prod_{y_2 \in \{0,1\}} \prod_{y_3 \in \{0,1\}} .... \sum_{y_k \in \{0,1\}} \widetilde{\phi}(y_1, ...y_k).$$

Basically all the $\exists$ is replaced by summations,the $\forall$ replaced by products and the formula ϕ is replaced by $\widetilde{\phi}$.

2.2 coNP $\subseteq$ IP

The language $\overline{\mathsf{SAT}} = \{\phi \mid \phi\ is\ not\ satisfiable\}$ is complete for the class coNP. It is enough to prove that $\overline{\mathsf{SAT}} \in \mathsf{IP}$. A formula

$$\phi is\ not\ satisfiable\ \Leftrightarrow \forall_{y_1} \forall_{y_2} ... \forall_{y_k} \phi(y_1, ..., y_k)\ is\ false.$$

By arithmetizing ϕ we get

$$\forall_{y_1} \forall_{y_2} ... \forall_{y_k} \phi(y_1, ..., y_k)\ is\ false \Leftrightarrow \prod y_1 \in \{0,1\} \prod y_2 \in \{0,1\} ..... \prod y_k \in \{0,1\} \widetilde{\phi}(y_1, ..., y_k) = 0.$$

In the above expression the degree of each variable in atmost nk. The Interactive Protocol for #SAT can be used here and the

$$Pr[Error] \leq \frac{nk^2}{|S|}.$$

The size of S can be chosen appropriately to bound the error. Hence coNP $\subseteq$ IP.

2.3 PH $\subseteq$ IP

We have the following characterisation for PH, For a $L \in$ PH,

$$x \in L \Leftrightarrow \exists_{y_1} \forall_{y_2} .....Q_{k y_k}(\phi(y_1, ..., y_k)) \Leftrightarrow \sum_{y_1 \in \{0,1\}} \prod_{y_2 \in \{0,1\}} ....\widetilde{\phi}(y_1, ..., y_k) \neq 0. \qquad (1)$$

Let l be the size of ϕ. Hence the degree of a variable in $\widetilde{\phi}$ is atmost l and in the whole expression is atmost $2^{k/2}l$ where k is the number of variables. The protocol given for SAT works and the

$$Pr[Error] \leq \frac{2^{k/2}lk}{|S|}.$$

It can be seen that the probalility is bound by a quantity which is exponentiol in k, but k being a constant, S can be chosen appropriately so the probalility of error is not high. Hence PH $\subseteq$ IP.

3 PSPACE $\subseteq$ IP

3.1 A characterization for PSPACE (Quantifier Boolean Formula)

Lemma 3. *For a language $L \in$ PSPACE $\exists B \in$ P such that*

$$x \in L \Leftrightarrow \exists_{y_1} \forall_{y_2} .....\exists_{y_{k-1}} \forall_{y_k}(x, y_1, y_2, ..., y_k) \in B$$

where $k = p(n)$ for a polynomial p.

Proof. Let $L \in$ PSPACE and $p(n)$ be the space used by the machine where p is a polynomial. Consider the configuration graph of the machine whose number of vertices equals $2^{p(n)}$ and the graph is known implicitly to us. We define the predicates using the configuration graph. Let α be the initial configuration and β be the final configuration. Now define the predicate $reach(\alpha, \beta, t)$ which is true if there is a path of length 2^t from α to β. So we have,

$$x \in L \Leftrightarrow reach(\alpha, \beta, t).$$

$$\begin{aligned} reach(\alpha, \beta, t) &\equiv \exists_\gamma (reach(\alpha, \gamma, t-1) \wedge reach(\gamma, \beta, t-1)) \\ &\equiv \exists_\gamma \forall_b ((reach(\delta, \delta', t-1)) \wedge (b = 0 \Rightarrow (\delta = \alpha \wedge \delta' = \gamma)) \wedge (b = 1 \Rightarrow (\delta = \gamma \wedge \delta' = \beta))) \end{aligned}$$

Now we recursively expand $reach(\delta, \delta', t-1)$ and by pushing all the expressions to the right yields

$$x \in L \Leftrightarrow \exists_{y_1} \forall_{y_2} .....\exists_{y_{k-1}} \forall_{y_k} \phi(y_1, y_2, ..., y_k) \; is \; satisfiable.$$

where $k = t = p(n)$. Hence the theorem □

Arithmetizing ϕ to $\widetilde{\phi}$ yields

$$x \in L \Leftrightarrow \sum_{y_1 \in \{0,1\}} \prod_{y_2 \in \{0,1\}} ....\widetilde{\phi}(y_1, ..., y_k) \neq 0.$$

3.2 Shamir's Interactive Protocol for PSPACE

The degree of a variable in the polynomial is atmost $2^{\frac{p(n)}{2}} l$ where is l is the degree of a variable in $\widetilde{\phi}$. The protocol used for #SAT yields a

$$Pr[Error] \leq \frac{2^{\frac{p(n)}{2}} l}{|S|}.$$

For $Pr[Error] \leq \frac{1}{2}$ the size of S has to be exponential which is not desired. The source of the degree blow up is due to $\frac{k}{2}$ products in the formula. Now we show a method to modify the formula in such a way that, for any variable there is atmost one $\forall$ quantifier between its apprarance and quantification. This will ensure that the degree of each variable is atmost $2l$ where l is the size of the formula.
Now given a formula

$$\exists_{y_1} \forall_{y_2} .....\exists_{y_{k-1}} \forall_{y_k} \phi(y_1, y_2, ..., y_k) \equiv \exists_{y_1} \forall_{y_2} \exists_{y_1'} (y_1 = y_1') .....\exists_{y_{k-1}} \forall_{y_k} \phi(y_1', y_2, ..., y_k)$$

The above modification to the formula makes sure that the degree of y_1 is atmost $2l$. But the new introduced variable y_1' has a high degree. We do this repeatedly until the degree of all the original variable and the newly introduced variables is atmost $2l$. By the above procedure we introduce atmost $\frac{k}{2}$ new variables for each original variable. The formula is still in polynomial in length. Now arithmetizing the new formula to obtain $\widetilde{\phi}$ which can be used in the protocol.
Now we hit a snag if we use the protocol used for #SAT. The reason being the value of the expression can be as large as 2^{2^l}. Now the 'Prover' cannot send the value of the expression as its size is exponential. In order to overcome the problem we look at $q \ mod \ n$ where q is the value of the expression and $n = 2^{2^l}$.
By the **The Chinese Remainder Theorem**, n can be expressed as $n = \prod_{i=1}^{k} p_i$ where $p_i's$ are prime or relatively prime and

$$x \equiv 0 \ mod \ n \Leftrightarrow ((x \equiv 0 \ mod \ p_1)...(x \equiv 0 \ mod \ p_k)).$$

In order to verify $q \neq 0$, it is sufficient to obtain a prime factor p of n from the 'Prover' for which $q \ mod \ p \neq 0$. Now it is enough for the 'Verifier' to check for the primality of p and the computations of the polynomials being done modulo p. The 'Prover' is asked to commit the value of the p at the start. Hence the new protocol works with $Pr[Error] \leq \frac{2kl}{|S|}$ which can be bounded by a desired value by appropriately choosing S.