6.841 Advanced Complexity Theory Mar 30, 2009

Lecture 14

Lecturer: Madhu Sudan *Scribe: Huy Nguyen*

1 Admin

The topics for today are:

- Interactive proofs
- The complexity classes IP and AM

Please see Madhu if you have not been assigned a project.

2 Theorems vs. Proofs

There is a long history of the notions theorems and proofs and the relation between them. The question about the meaning of these notions is implicit in Hilbert's program, where he asked if you could prove theorems in various general contexts. Then in Godel's work, he proved that no logic system can be both complete and consistent. The notion of P and NP came along also from the investigation of this relation, as evident in the title of Cook's paper "The complexity of theorem-proving procedures" [2].

In the early works, a system of logic consists of a set of axioms and the derivation rules. A theorem is just a string of characters. The axioms are the initial true statements and the derivation rules show how to get new true statements from existing ones. A **proof** is a sequence of strings where each string is a either an axiom or derived from previous ones by derivation rules. The final string of the proof should be the derived **theorem**.

In computational complexity, we abstract this procedure away. Theorems are statements that have proofs such that the pair (theorem, proof) is **easy** to verify. With this abstraction, we have separated the theorem from the proof. Intuitively, the complexity class P is roughly equivalent to the complexity of verifying proofs, while the complexity class NP is roughly equivalent to the complexity of finding proofs. In a sense, computational complexity is the study of the relation between theorems and proofs.

3 Interactive Proof

In cryptography, there are related problems. You want to be able to prove that **you** are authorized to perform some **action** e.g. you to want to prove you are allowed to access your bank account and withdraw money, etc. However, someone else who somehow obtained the entire transcript of your transaction cannot repeat the action. In other words, the proof should not be **replayable**. This is completely different from the conventional settings. Thus it is unclear if non-replayable proofs are achievable.

Goldwasser, Micali, and Rackoff [4] answered this question positively and introduced several new notions. The first one is interactive proof. In order to prove a theorem T, the prover interacts with the verifier as follows. The prover sends the proof π to the verifier. The verifier finds some part of the proof unclear and sends a question q_1 to the prover. The prover sends back an answer a_1. Then the verifier sends follow up questions and the prover sends back answers. After k rounds of interaction, the verifier tosses some random coins R and runs a probabilistic polynomial time procedure $V(T, \pi, a_1, \ldots, a_k, R)$, which returns 1 whp if T is true, and 0 whp if T is false. The second new notion is the knowledge complexity of interaction. They showed that in various context, there exists zero knowledge proofs. This means that after seeing the proof, the verifier knows the theorem is true and things it already knows but nothing else. For example, after seeing a proof that a graph is Hamiltonian, the verifier still does not know if an edge is in the Hamiltonian circuit or not. Interestingly, interactive proofs not only allow zero knowledge proofs but also proving more things.

IP is defined to be the class of languages that admit interactive proofs with probabilistic polynomial time verifier. In other words, $L \in IP$ iff $\exists$ ppt V such that $\forall x \in \{0,1\}^n$, on input x, V interacts with the prover P and terminates in polynomial time with 0/1 verdict:

- If $x \in L$ then $\exists P(x)$ such that $V(x) \leftrightarrow P(x)$ accepts with probability at least $\frac{2}{3}$.
- If $x \in L$ then $\forall P^*(x)$, $V(x) \leftrightarrow P^*(x)$ accepts with probability at most $\frac{1}{3}$.

Let $IP[k]$ denote the set of languages with interactive proofs using k rounds of interactions.

4 Arthur-Merlin Proofs

Babai and Moran [1] came up with a notion of Arthur-Merlin (AM) Proofs, which seems to be very close to NP but they were unable to prove they are equal. They were interested in some number theoretic problems L that they think are in P but do not know how to prove. L is in NP but they do not think L is NP-complete. They were unable to prove that $\overline{L} \in NP$ but they proved $\overline{L} \in AM$.

An AM proof goes as follows. To decide if x is in a language L, Arthur sends a question q to Merlin, and Merlin sends back an answer a. Arthur then runs a verification procedure $V(x, q, a)$, which returns a 0/1 verdict. This protocol is the same as IP except that Babai and Moran are interested in 2 round interaction and Arthur does not have any hidden information. Intuitively, Arthur is more naive (he trusts Merlin) so Merlin can cheat. Therefore, there are fewer languages remaining sound.

5 Graph non-isomorphism

Define graph non-isomorphism (GNI) to be the language of pairs of graphs (G_0, G_1) such that G_0 and G_1 are non-isomorphic. We will now show a private coin, two round, interactive proof for graph non-isomorphism due to Goldreich, Micali, and Wigderson [3]. Suppose that the graphs' vertices are $[n]$. The protocol is as follows.

1. The verifier V picks a random permutation $\pi : [n] \to [n]$ and a bit $b \in \{0,1\}$. V then sends $H = \pi(G_b)$ to the prover P.
2. If G_0 and G_1 are non-isomorphic, H check if $H = \pi(G_0)$ for some π. If so, it sends $c = 0$ back to V. Otherwise, it sends $c = 1$.
3. The verifier V accepts if $b = c$ and rejects otherwise.

If the graphs are isomorphic, the distributions of graphs being sent by V are the same regardless of b so the prover can only guess right with probability $\frac{1}{2}$. If they are not isomorphic, the prover can always get the answer right. That proves $GNI \in IP[2]$. Note that this protocol relies crucially on the private information (the bit b). Surprisingly enough, interactive proofs are not inherently dependent on private information. Any interactive proof can be modified to use public random coins only while preserving the number of rounds.

6 Historical issues of interactive proofs

There are three interesting issues of interactive proofs:

1. Public coin vs. private coin.
2. Number of rounds. The interesting values are constant and polynomial.
3. One-sided error vs. two-sided error (similar to RP versus BPP)

Interestingly, both the first and the third does not matter: the two choices are equivalent. Any protocol can modified to use public coins and get one-sided error while preserving the number of rounds.

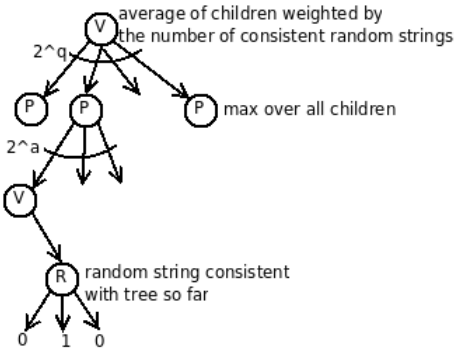

Figure 1: An interaction tree and the way to compute the maximum acceptance probability at each node given the probabilities at its children

7 $IP \subseteq PSPACE$ [4]

Consider a language L in IP and an input $x \in \{0,1\}^n$. The question is whether we can compute the maximum acceptance probability of V over all provers P. If this can be done in $PSPACE$ then the proof immediately follows. Let's look at the interaction tree of the protocol in figure 1.

When we compute this tree bottom up, at every level, it is easy to tell what is the best strategy for the prover. This computation can be done in $PSPACE$. Therefore, $IP \subseteq PSPACE$.

8 Public coin one-sided error protocol with polynomial number of rounds

Now we show that every protocol can be modified to use public coin and get one-sided error using polynomially many rounds of interaction. The proof is due to Kilian [**?**].

Assume that all questions and answers are either 0 or 1. Rather than executing the interaction tree, we use a new prover that tells how many accepting paths there are in the tree.

The protocol goes as follows. The prover tells the verifier the number of accepting paths from the current nodes N' and from each of its children N'_0 and N'_1. Let the corresponding real numbers be N, N_0, N_1, respectively. The verifier checks if $N' = N'_0 + N'_1$. Then it moves to the bth child with probability N'_b/N'. This is repeated until they get to a leaf. The verifier accepts if it is an accepting leaf and the number of paths from the root of the tree (given by the prover) is at least 2/3 the total number of random strings.

Claim 1 *The probability that the verifier accepts is at most* $\frac{N}{N'}$.

Proof Note that the protocol uses no private randomness. With an honest prover, the probability that the verifier accepts is $1 = \frac{N}{N'}$ when $x \in L$. When $x \notin L$, in order to cheat, the prover needs to give N' much larger than N. We can prove the claim using induction. The probability that the verifier accepts is at most $\frac{N'_0}{N'} \cdot \frac{N_0}{N'_0} + \frac{N'_1}{N'} \cdot \frac{N_1}{N'_1} = \frac{N_0}{N'} + \frac{N_1}{N'} = \frac{N}{N'}$. Thus, the prover cannot cheat by giving N' much larger than N. ■

Next time we will show a one sided error public coin $O(k)$ round protocol for $IP[k]$. This result uses the protocol given by Goldwasser and Sipser [5] that can proves a set $S \subset \{0,1\}^n$ is "large" if membership in S is in $AM[k]$. More precisely, if $|S| \geq f(n)$ then the protocol accepts with probability 1. If $|S| \leq \frac{f(n)}{n^2}$, it accepts with probability at most $\frac{1}{2}$.

For example, consider the GNI protocol. In the protocol we showed, the random coin R can only be revealed after P committed an answer. Using the above protocol, the prover can show that given its choice, the set of accepting leaves is large. Then the protocol can be repeated another time to show the set of nodes with many accepting leaves is large. The details will be shown next time.

References

[1] László Babai and Shlomo Moran. Arthur-merlin games: a randomized proof system, and a hierarchy of complexity class. *J. Comput. Syst. Sci.*, 36(2):254–276, 1988.

[2] Stephen A. Cook. The complexity of theorem-proving procedures. In *STOC '71: Proceedings of the third annual ACM symposium on Theory of computing*, pages 151–158, New York, NY, USA, 1971. ACM.

[3] Oded Goldreich, Silvio Micali, and Avi Wigderson. Proofs that yield nothing but their validity or all languages in np have zero-knowledge proof systems. *J. ACM*, 38(3):690–728, 1991.

[4] S Goldwasser, S Micali, and C Rackoff. The knowledge complexity of interactive proof-systems. In *STOC '85: Proceedings of the seventeenth annual ACM symposium on Theory of computing*, pages 291–304, New York, NY, USA, 1985. ACM.

[5] S Goldwasser and M Sipser. Private coins versus public coins in interactive proof systems. In *STOC '86: Proceedings of the eighteenth annual ACM symposium on Theory of computing*, pages 59–68, New York, NY, USA, 1986. ACM.