Lecture 11. Interactive Proof Systems

We consider another generalization of the NP-computation, "interactive proof system" or "interactive proof" (in short, IP).

11.1 Interactive proofs

We begin with the standard one introduced by Goldwasser, Micali, and Racoff (in 1985). An *interactive proof* is a pair $(\mathtt{P}, \mathtt{V})$ of an "oracle machine" $\mathtt{P}$ and a polynomial-time randomized Turing machine $\mathtt{V}$ that share an input tape and a message tape. We call $\mathtt{P}$ a *prover* and $\mathtt{V}$ a *verifier*. The input tape contains an *input* given as a binary string, and the message tape contains a *message* that is given either by the prover or the verifier alternatively. For bounding the computation time, we consider two polynomials p and r. For any input of length ℓ, we assume that the computation terminates with the verifier entering an accepting state or a rejecting state after exchanging messages for $r(\ell)$-rounds. We also assume that at each verifier's turn $\mathtt{V}$ runs within $p(\ell)$-time to read/write a message on the message tape.

Remark. Although we treat $\mathtt{P}$ as a machine for our explanation, it is simply a function that yields a binary string that is written as a message on the message tape based on the input and the exchanged messages, and it is not necessary to describe $\mathtt{P}$ as a machine. A prover can be randomized, in which case the prover has its own (private) random tape, and its outputs (i.e., its messages) may vary depending on random bits on its random tape. But for a while, let us consider only deterministic provers.

Intuitively, a prover is a generalized way to give a witness and a verifier is a generalization of a witness checking predicate for NP. For a formal definition, let us introduce the following probability, which we will call[1] the *convincing probability* of a prover $\mathtt{P}$.

$$\mathrm{Pconv}(\mathtt{P}; x) \;=\; \Pr_{\mathtt{V}}[\,\mathtt{V}(x) \text{ enters an accepting state after exchanging messages with } \mathtt{P}\,].$$

Definition 11.1 A decision problem L has an interactive proof if there exists a polynomial-time verifier $\mathtt{V}$ such that for all input x, we have

$$\begin{aligned} x \in L &\;\Rightarrow\; \exists \mathtt{P}\,[\,\mathrm{Pconv}(\mathtt{P}; x) > 2/3\,], \quad \text{and} \\ x \notin L &\;\Rightarrow\; \forall \mathtt{P}'\,[\,\mathrm{Pconv}(\mathtt{P}'; x) < 1/3\,]. \end{aligned}$$

More specifically, an interactive proof $(\mathtt{P}, \mathtt{V})$ is $r(\ell)$-round from a verifier's turn (resp., a prover's turn) if the communication starts from $\mathtt{V}$ (resp., $\mathtt{P}$) sending a message, and it ends after $r(\ell)$-rounds for any input of length ℓ.

For any polynomial $r(\cdot)$, A class $\mathrm{IP}[r(\ell)]$ is the class of decision problems with an $r(\ell)$-round interactive proof. In general, we use $\mathrm{IP}[\mathrm{poly}]$ to denote $\cup_{r:\mathrm{polynomial}}\mathrm{IP}[r(\ell)]$.

[1] This term is only for our class. In fact, I am not sure whether it is appropriate in English.

Remark. As a special case, we may consider an interactive proof satisfying the following condition for L.

$$\begin{array}{rcl} x \in L & \Rightarrow & \exists \mathtt{P}\,[\,\mathrm{Pconv}(\mathtt{P}; x) = 1\,], \quad \text{and} \\ x \notin L & \Rightarrow & \forall \mathtt{P}'\,[\,\mathrm{Pconv}(\mathtt{P}'; x) < 1\,]. \end{array}$$

A problem with this type of interactive proof (and an $r(\ell)$-round interactive proof) is nothing but a problem in the class $\mathrm{ADepth}(r(\ell))$ that was discussed in the last lecture. Thus, PH can be defined as the class of problems with a constant-round interactive proof of this type, and NP can be defined as the class of problems with a one-round interactive proof of this type that starts from a prover's turn.

Let us see some example of an interactive proof. Consider the following problem.

Graph Isomorphism Problem (GI)
Instance: Two undirected graphs G_1 and G_2 (with n vertices).
Question: Are they isomorphic (i.e., equivalent) graphs?

Note that we are given graphs in the following way. Thus, the problem may not be so easy.

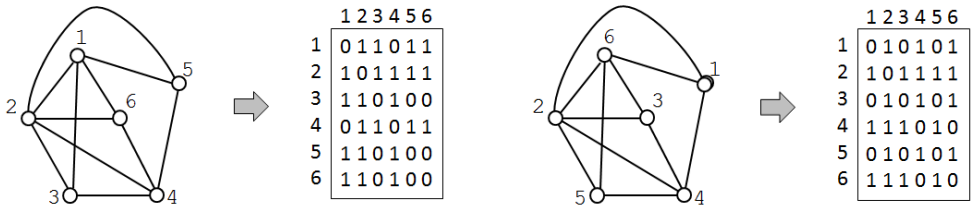

Figure 10.1 Two isomorphic graphs and their codings

An *isomorphism* (between two graphs) is a permutation ϕ from $\{1, 2, \ldots, n\}$ to $\{1, 2, \ldots, n\}$ indicating which vertex in one graph corresponds to a given vertex in another graph. Given a permutation candidate it is easy to check its correctness. Furthermore, each permutation is expressed by n input-output pairs. Thus, GI is an NP problem. On the other hand, it has not known whether GI is in P or not.

Since GI is an NP problem, it is easy to see that the following problem is in coNP.

Graph Non-Isomorphism Problem (GNI)
Instance: Two undirected graphs G_1 and G_2 (with n vertices).
Question: Are they non isomorphic?

Again it seems hard to give a standard witness for a given pair of graphs are not isomorphic. Interestingly, we can give such a witness quite easily in the interactive system framework.

Theorem 11.1 GNI has an 2-round interactive proof starting from a verifier's turn. That is, GNI $\in$ IP[2].

11.2 Interactive proof classes and their relations

Interactive proof systems have been introduced independently by several researchers from two different motivations, one for generalizing the notion of NP to understand the nature of NP-computation, and another for applications in cybersecurity systems. Then around 1990, many interesting results have been obtained on the power of interactive proofs. Here we review some of them quickly. See textbooks in the reference list of this handout for further reading.

In an interactive proof, a verifier can make use of a "private" random sequence, which was an important role for designing the above interactive proof for GNI. Babai (in 1985) introduced an interactive proof system where the verifier always make its random sequence public (to the prover) by writing it down on the message tape, which he called an *Arthur-Merlin protocol* (where Arthur is a verifier and Merlin is a prover). A protocol is called AM[k] if it starts from an Arthur's round and terminates within k rounds. On the other hand, MA[k] is a protocol starting from an Merlin's round and terminates in k rounds. (Note that in an MA[2] protocol, the last verifier's message is not used, but it is used for the verfier to check the correctness of the "witness" given by the prover.)

We use AM, etc., to denote the class of decision problems with the corresponding AM-type protocol. The following relation is known. (The proof of NP $\subseteq$ MA[2] is immediate, while we need some combinatorial technique to show MA[2] $\subseteq$ AM[2].)

Theorem 11.2 NP $\subseteq$ MA[2] $\subseteq$ AM[2].

We have the following interesting/amazing results on AM and IP classes.

Theorem 11.3 (Babai '85, Babai and Moran '88)
For any $k \geq 2$, AM[k] $\subseteq$ AM[2].

Theorem 11.4 (Goldwasser-Sipser '86)
For any $r(\cdot)$, IP[$r(\ell)$] $\subseteq$ AM[$r(\ell) + 2$].

Theorem 11.4 (Shamir '92)
IP[poly] = AM[poly] = PSPACE.

A very important variant of "interactive proof" is a "zero-knowledge interactive proof" (in short, ZKIP), which was introduced by Goldwasser, Micali, and Rackoff in 1985 (in the same paper that the notion of interactive proof was introduced). Roughly speaking, an interactive proof is called a *zero-knowledge interactive proof* if any observer of messages exchanged by the verifier and the prover on any positive input cannot get any information except that the input is indeed positive. We can easily check that our interactive proof for the GNI problem is indeed zero-knowledge. We also see, e.g., the 3Coloring problem, has an interactive proof. Researchers propose to use such a ZKIP for a good identification method.

References:

1. M. Sipser, *Introduction to the Theory of Computation* (3rd edition), Cengage Learning, 2013, ISBN-13: 9781133187790
2. D. Du and K. Ko, *Theory of Computational Complexity* (2nd edition), John Wiley and Sons, Inc., 2000, ISBN:978-1-118-30608-6.
3. O. Goldreich, *Foundations of Cryptography: Vol.1 Basic Tools*, Cambridge, 2001.

Homework exercise from Lecture 11

Homework rule: From this lecture, I will give you some advanced problems for which you need to do some literature survey to solve them. Choose one of these problems (from six problems I will give from this and next two lectures) and submit your report by Nov. 24th (Friday) noon to Watanabe's mail box in the mail box room of the West 8E building. You can submit your report by email. You can get 3 points by submitting an OK report. Please do not solve more than one problem.

For these problems, you can find enough information from web pages. Be careful to choose a reliable source. You have to understand what is written there and write its digest in two to three pages by yourself. Of course, the information on the source of your explanation should be given in the reference list of your report.

For writing an answer, you may use Japanese.

Advanced problems

1. Show that GNI is in AM[2].
2. There is a variation of ZKIP that can be used for an identification system. Explain it with an appropriate example.