23 Worksheet for Lecture 23 (Limitations of IPs)

Problem 23.1 (IPs with small-space verifier)**.** In this question you will prove that the languages that have laconic IPs where the verifier runs in small space are easy to decide.

1. Suppose that a language $\mathcal{L}$ has a *public-coin* interactive proof with completeness error ϵ_{c} and soundness error ϵ_{s} such that $\epsilon_{\mathsf{c}} + \epsilon_{\mathsf{s}} < 1$ and the communication and verifier space complexity are upper bounded by s. Prove that $\mathcal{L}$ can be decided by an algorithm running in space $O(s^2)$.

2. Prove the same without the public coin assumption.

This simple observation has an important consequence: since it is believed that $\mathsf{DTIME}[T] \not\subseteq \mathsf{SPACE}[o(T)]$, we should not expect every language in $\mathsf{DTIME}[T]$ to have an interactive proof with $o(\sqrt{T})$ verifier time. This is different from the case for PCPs, where we have constructions for every language in $\mathsf{NTIME}[T]$ with $\mathsf{polylog}(T)$ verifier time.

Problem 23.2 (laconic provers with perfect completeness)**.** Suppose that a language $\mathcal{L}$ has a proof system with perfect completeness and non-trivial soundness, in which the prover-to-verifier communication is at most c_{p} bits. Show that $\mathcal{L} \in \mathsf{coNTIME}(2^{c_{\mathsf{p}}(n)} \cdot \mathsf{poly}(n))$.

You may use Zermelo's Theorem: "in every finite, deterministic, perfect-information game between players A and B where the players move alternately, if the game cannot end in a draw, either A or B has a winning strategy."

(*Hint: Use the IP (P, V) to define a two-player game: A corresponds to P, and the goal of B is to generate an interaction that makes V reject.*)

Problem 23.3 (IPs with laconic prover)**.** Suppose that a language $\mathcal{L}$ has a one-round interactive proof where the prover sends c_{p} bits: the verifier sends a message, then the prover sends a c_{p}-bit message, and finally the verifier decides based on the prover's message and its internal randomness. (For example, GNI has such an interactive proof.)

We shall prove that $\mathcal{L} \in \mathsf{BPTIME}[2^{O(c_{\mathsf{p}} \log c_{\mathsf{p}})} \cdot \mathsf{poly}(n)]^{\mathsf{NP}}$.

1. We first consider the task of sampling a random witness uniformly at random for an NP language with a small number of witnesses. Suppose that the relation $\mathcal{R} \subseteq \{0,1\}^* \times \{0,1\}^*$ is a relation for an NP language such that instances have at most $O(n)$ different witnesses. In other words, if an n-bit string $\mathbb{x}$ is such that $(\mathbb{x}, \mathbb{w}') \in \mathcal{R}$ for some NP-witness $\mathbb{w}'$, then the set $\mathcal{R}_{\mathbb{x}} \coloneqq \{\, \mathbb{w} : (\mathbb{x}, \mathbb{w}) \in \mathcal{R} \,\}$ satisfies $|\mathcal{R}_{\mathbb{x}}| = O(n)$. Show that there exists a probabilistic polynomial-time machine with an NP oracle which outputs a uniformly random $\mathbb{w} \in \mathcal{R}_{\mathbb{x}}$ given the input $\mathbb{x}$. *Hint: Consider the languages*

$$\mathcal{L}_1 \coloneqq \{\, (\mathbb{x}', 1^k) : \exists \textit{ distinct } \mathbb{w}^{(i)} \textit{ s.t. } (\mathbb{x}', \mathbb{w}^{(i)}) \in \mathcal{R}, \forall i \in [k] \,\} \quad ,$$
$$\mathcal{L}_2 \coloneqq \{\, (\mathbb{x}, 1^k, 1^i, 1^j) : \exists \textit{ distinct } \mathbb{w}^{(l)} \textit{ s.t. } (\mathbb{x}, \mathbb{w}^{(l)}) \in \mathcal{R}, \forall l \in [k], \textit{ and } \mathbb{w}^{(i)}_j = 0 \,\} \quad ,$$

   and show that $\mathcal{L}_1, \mathcal{L}_2 \in \mathsf{NP}$.

2. Reduce the general case, where $\mathcal{R}_{\mathbb{x}}$ is an arbitrarily large set of $\mathsf{poly}(n)$-size bit strings, to the previous one. You may use the following fact without proof: "For every $a, b \in \mathbb{N}$ with $b > a$, there exists a family of *hash functions* $\mathcal{H} \subseteq \{\, h \colon [b] \to [a] \,\}$ satisfying the following condition.

When $h \in \mathcal{H}$ is picked uniformly at random, with high probability $\left|h^{-1}(c)\right| = \Theta(b/a)$ for all $c \in [a]$ simultaneously. Furthermore, such an h can be sampled in probabilistic polynomial time."

3. Combine the results above with the approach for public-coin interactive proofs shown in class to prove that $\mathcal{L} \in \mathsf{BPTIME}[2^{O(c_{\mathsf{P}} \log c_{\mathsf{P}})} \cdot \mathsf{poly}(n)]^{\mathsf{NP}}$.

24 Worksheet for Lecture 24 (Limitations of PCPs and IOPs)

Problem 24.1 (lower bound on soundness error)**.** Suppose that there exists $\mathbb{x} \notin \mathcal{L}$ such that for every choice of verifier randomness $\rho \in \{0,1\}^{\mathsf{r}}$ there exists a proof $\pi \in \Sigma^{\mathsf{l}}$ such that $V^{\pi}(\mathbb{x};\rho) = 1$. Prove that $\epsilon \geq 2^{-\mathsf{q}\log|\Sigma|}$.

Problem 24.2 (more on lower bounds)**.** The *Exponential Time Hypothesis* (ETH) states that 3SAT cannot be decided by any deterministic algorithm running in time $2^{o(n)}$. Prove that, assuming ETH, if $\mathcal{L} = \text{3SAT}$ has a PCP with perfect completeness such that $\mathsf{r} + \mathsf{q}\log|\Sigma| = o(n)$, then $\epsilon \geq 2^{-\mathsf{q}\log|\Sigma|}$. (*Hint: prove that ETH implies the assumption to the prior problem.*)

Problem 24.3. In class, we have seen that non-adaptive two-query PCPs over the binary alphabet are unlikely by reducing the verifier decision to a CSP with binary alphabet and arity 2. In this problem, we will generalize the idea of using CSP solvers to decide the language.

Let $\mathsf{PCP} = (P, V)$ be a non-adaptive PCP for n-variable 3SAT with perfect completeness, soundness error ϵ_{s}, alphabet Σ, proof length $\mathsf{l} = 2^{o(n)}$, query complexity q, randomness complexity $\mathsf{r} = o(n)$, and verifier running time $t_{\mathsf{v}} = 2^{o(n)}$. Let $\mathcal{S}$ be a deterministic algorithm that decides in time $\mathsf{poly}|\psi|$ whether a CSP instance ψ over alphabet Σ with arity q has value 1 or value at most ϵ_{s}.

1. Show that there exists a $\mathsf{poly}(2^{\mathsf{r}}, t_{\mathsf{v}}, \mathsf{l})$-time reduction that takes an instance $\mathbb{x}$ and outputs a CSP instance ψ over the alphabet Σ with arity q and size $\mathsf{poly}(2^{\mathsf{r}}, t_{\mathsf{v}}, \mathsf{l})$ such that if $\mathbb{x} \in \mathcal{L}$, then the value of ψ is 1 and if $\mathbb{x} \notin \mathcal{L}$, then the value of ψ is at most ϵ_{s}.

2. Show that there exists an algorithm $\mathcal{S}'$ that decides whether a 3SAT formula φ over n variables is satisfiable in time $2^{o(n)}$.

3. Use the fact that satisfiability of a CSP with binary alphabet and arity 2 can be decided in polynomial time to deduce that non-adaptive two-query PCPs over the binary alphabet are unlikely.[2] In particular, show that if $\epsilon_{\mathsf{s}} < 1$ and $\mathsf{q} = 2$, then 3SAT can be decided in time $2^{o(n)}$.

4. Use the fact that there exists a polynomial-time algorithm that decides whether a CSP instance with binary alphabet and arity 3 has value 1 or at most 5/8 to deduce that non-adaptive three-query PCPs over the binary alphabet with soundness error 5/8 are unlikely.[3] In particular, show that if $\epsilon_{\mathsf{s}} < 5/8$ and $\mathsf{q} = 3$, then 3SAT can be decided in time $2^{o(n)}$.

[2] It follows from Schaefer's dichotomy theorem.
[3] It is called Zwick's algorithm.