Advanced Proof-Systems	Winter 2022 - 2023
Problem Set 4	
Lecturer: Ron Rothblum	Due: 18/1/2021 at 4pm

In this exercise we will see how to use the doubly-efficient interactive proof for low depth circuits that we developed in class, to obtain powerful results (that are not directly related to circuits).

1 $\mathsf{PSPACE} \subseteq \mathsf{IP}$

In this exercise we will prove that $\mathsf{PSPACE} \subseteq \mathsf{IP}$ (the more interesting, and difficult, part of the $\mathsf{IP} = \mathsf{PSPACE}$ theorem). Recall that this means that a huge class of computations can be proved (e.g., given a particular setting of a chessboard, that white has a winning strategy).

The proof that we will give here is not the "traditional" proof, but actually gives better parameters.

1. Consider the problem of ST-connectivity. In this problem the input is a directed graph $G = (V, E)$ and two vertices $s, t \in V$. The goal is to decide whether there is a directed path from s to t.

   Show that ST-connectivity can be solved in NC_2 (i.e., by a poly($|V|$)-size and $O(\log^2(|V|))$ depth circuit).

   Guideline: Given a graph $G = (V, E)$, and an integer i, define $A_G^{(i)}$ to be a $|V| \times |V|$ matrix whose (u, v)-th entry is 1 if there is a path from u to v of length at most i, and 0 otherwise.

   First show an $O(\log(|V|))$-depth poly($|V|$)-size circuit that given as input $A_G^{(i)}$ computes $A_G^{(2i)}$.

2. Consider a variant of ST-connectivity in which the input graph (V, E) can have huge size but is represented implicitly as follows: rather than an explicit description of the graph, the input is 1^n (i.e. the number n in unary) and a poly(n)-time Turing machine M. The vertex set is $V = \{0, 1\}^n$ and the edges are $E = \{(u, v) \in V^2 \mid M(u, v) = 1\}$. Note that this allows us to represent an exponential size graph using a polynomial-time algorithm.

   Let $\mathcal{L} \in \mathsf{PSPACE}$. Show a (Karp) reduction from $\mathcal{L}$ to this variant of ST-connectivity.

   Note: A detailed description suffices. There is no need to go into annoying Turing machine details.

3. Conclude that every problem in PSPACE can be solved by an exponential-size polynomial-depth circuit.

4. Show that $\mathsf{PSPACE} \subseteq \mathsf{IP}$. For this exercise you are allowed to assume that the doubly-efficient interactive proof for (logspace-uniform) NC that we saw in class works for *any* size S and depth D circuit, with verification time $(D + n) \cdot \text{polylog}(S)$.[1]

[1] This is not exactly true, as it only works for *uniform* circuits. We ignore this subtlety here.

5. Suppose that language L can be solved by a Turing Machine in time T and space S. The above protocol is an interactive proof for this language in which the verifier's runs in roughly S time. What is the complexity of the prover?

   Food for thought (and major open question): what running time for the honest prover would you hope to achieve?

2 Batch Verification for P

Suppose that we want to check whether k (potentially unrelated) inputs $x_1, \ldots, x_k$ all belong to a language $\mathcal{L}$, where $\mathcal{L}$ is a class for which membership of a *single* input x can be decided in time $t = \mathrm{poly}(n)$. Clearly this task can be accomplished in time $k \cdot t$. The goal of this exercise is to show that membership of all of these k inputs in $\mathcal{L}$ can be *verified* (via a doubly efficient interactive proof) in time roughly proportional to $t + \log(k)$. That is, the cost of verifying k inputs is more or less like the cost of computing just one.

More formally, let $\mathcal{L}$ be a language that is computable in time $t = \mathrm{poly}(n)$ and let $k = \mathrm{poly}(n)$ be a parameter. Consider the language:

$$\mathcal{L}' = \Big\{ (x_1, \ldots, x_k) : \forall i \in [k] \; x_i \in \mathcal{L} \text{ and } |x_i| = |x_1| \Big\}$$

(i.e., $\mathcal{L}'$ consists of tuples of k equal length strings all of which belong to $\mathcal{L}$).

Show that there exists an interactive proof for $\mathcal{L}'$ so that on input $(x_1, \ldots, x_k) \in (\{0,1\}^n)^k$, the verifier runs in time $\tilde{O}(n \cdot k) + (\log(k) + t) \cdot \mathrm{polylog}(n)$, the communication complexity is $(\log(k) + t) \cdot \mathrm{polylog}(n)$ and the prover runs in time $\mathrm{poly}(n, k, t)$.

Assumption (+Hint): You may assume that $\mathcal{L}$ is accepted by a family of logspace uniform size $\tilde{O}(t)$ circuits.