Theory of Computation

Final Examination on December 23, 2022
Fall Semester, 2022

Problem 1 (20 points) Prove that if **NP** $\subseteq$ **ZPP**, then **NP** $\subseteq$ **BPP**. (Recall that a language in **ZPP** has two Monte Carlo algorithms, one with no false positives and the other with no false negatives. The class **BPP** contains all languages L for which there is a precise polynomial-time NTM N such that if $x \in L$, then at least 2/3 of the computation paths of N on x lead to "yes"; otherwise, at least 2/3 of the computation paths of N on x lead to "no.")

Proof: Assume **NP** $\subseteq$ **ZPP**. Pick any NP-complete language L. We only need to show that $L \in$ **BPP**. There exists a Las Vegas algorithm A that decides L in expected polynomial time, say $p(n)$. By Markov's inequality, the probability that the running time of A exceeds $3\,p(n)$ is at most 1/3. Run A for $3\,p(n)$ steps to determine with probability at least $1 - 1/3 = 2/3$ whether the input belongs in L. We therefore obtain a polynomial-time algorithm for L which errs with probability at most 1/3 on each input. Hence L is in **BPP**. ■

Problem 2 (20 points) **PSPACE** is the set of all languages which can be decided by a deterministic TM using polynomial space. Prove that **BPP** $\subseteq$ **PSPACE**.

Proof: Let M be a randomized polynomial-time TM that recognizes $L \in$ **BPP** with two-sided error-probability $\varepsilon \leq 1/4$. Let $r(n)$ be the number of coin tosses of M. Then TM decides L as follows. Count of the number s of accepting paths. If $s \geq (1-\varepsilon)2^{r(n)}$, then accept; otherwise, reject. By recycling space across executions of the loop in counting the number of accepting paths, this can be implemented in polynomial space. ■

Problem 3 (20 points) Let $G = (V, E)$ be an undirected graph in which every node has a degree of at most k. Let I be a nonempty set. I is said to be independent if there is no edge between any two nodes in I. MAXIMUM INDEPENDENT SET finds the largest independent set in G. Consider the greedy following algorithm for MAXIMUM INDEPENDENT SET:

Algorithm 1

$I := \phi$;
while $\exists\, v \in G$ **do**
    Add v to I;
    Delete v and all of its adjacent nodes from G;
end while
return I;

Prove that this algorithm for MAXIMUM INDEPENDENT SET is a $\frac{k}{k+1}$-approximation algorithm. Recall that an ε-approximation algorithm returns a solution that is at least $1 - \varepsilon$ times the maximum solution.

Proof: Since each stage of the algorithm adds a node to I and deletes at most $k+1$ nodes from G, I has at least $\frac{|V|}{k+1}$ nodes, which is at least $\frac{1}{k+1}$ times the size of the maximum independent set because the size of the maximum independent set is trivially at most $|V|$. Thus this algorithm returns solutions that are never smaller than $1 - \frac{1}{k+1} = \frac{k}{k+1}$ times the maximum. ■

Problem 4 (20 points) Let C_n be a boolean circuit which has n boolean inputs. Language $L \subseteq \{0,1\}^*$ has polynomial circuits if there is a family of circuits $\mathcal{C} = (C_0, C_1, \ldots)$ such that C_n accepts $L \cap \{0,1\}^n$ and the size of C_n is at most $p(n)$ for some fixed polynomial p. Prove or disprove that **IP** contains all languages that have polynomial circuits.

Proof: No. Polynomial circuits can accept undecidable languages which are clearly not in **IP**. See p. 268 of the textbook. ■

Problem 5 (20 points) #HAMILTONIAN PATH computes the number of Hamiltonian paths in a graph. Prove that #HAMILTONIAN PATH is in #**P**.

Proof: Let $f(G)$ be the number of Hamiltonian paths of the input graph G. A polynomial-time NTM M guesses a path on G and accepts it if the path is Hamiltonian. Then $M(G)$ has $f(G)$ accepting paths for all input graphs G. So $f \in \#\mathbf{P}$. ■