COMS 4236 – Homework 2

DUE ON GRADESCOPE: Oct 15, 11:59pm

Instructions. Please review the course **Syllabus** for policies on collaboration and use of AI before starting this assignment. Please submit a typed (not hand-written) pdf of your solutions on Gradescope, with one question per page.

1. Does rereading a certificate change its power?

A deterministic verifier has a read-only input x, a separate read-only certificate y of length at most $p(n)$, and $O(\log n)$ work space. Here p is a fixed polynomial and $n = |x|$. It halts in polynomial time on every pair (x, y). A string x is accepted when *some* allowed certificate makes the verifier accept.

(a) Suppose the certificate head can stay put or move right, but cannot move left. Prove that the languages with such verifiers are exactly NL.

(b) Now allow the certificate head to move both left and right. Prove that the resulting class is NP. Your argument may use a certificate that records an entire polynomial-length accepting computation, rather than just its choices.

2. A space simulation gives a time lower bound

Using Williams's theorem and the deterministic space hierarchy theorem, prove that there is a language A such that $A \in \mathsf{DSPACE}(n)$ but A is not in $\mathsf{DTIME}(n^{2-\varepsilon})$ for any fixed $0 < \varepsilon < 1$.

3. Borrowing registers that are not empty

All registers hold b-bit strings. Write $+$ for bitwise XOR and $\odot$ for bitwise AND. The input determines two strings x, y. Suppose recursive programs P_x, P_y have these net effects for *every* starting register contents:

$$P_x : (R_x, R_y, R_o) \mapsto (R_x + x, R_y, R_o), \qquad P_y : (R_x, R_y, R_o) \mapsto (R_x, R_y + y, R_o).$$

They may temporarily use all three registers but restore the other two when they return. Consider the following program:

$$\begin{array}{ll} R_o \leftarrow R_o + (R_x \odot R_y) & \text{call } P_x \\ R_o \leftarrow R_o + (R_x \odot R_y) & \text{call } P_y \\ R_o \leftarrow R_o + (R_x \odot R_y) & \text{call } P_x \\ R_o \leftarrow R_o + (R_x \odot R_y) & \text{call } P_y \end{array}$$

Read each row from left to right, then continue to the next row.

(a) Prove that the final effect is $(R_x, R_y, R_o) \mapsto (R_x, R_y, R_o + (x \odot y))$. Your proof must allow arbitrary initial contents; no initially zero register may be assumed.

(b) Using this product procedure and child programs that add their values into a chosen register, describe a clean-addition procedure for bitwise OR.

Explain why the recursive calls do not require fresh data registers. Finally, briefly explain what more you would need to show in order to prove that a uniform tree-evaluation algorithm uses only 3b bits of total space.

4. Estimating the fraction of satisfying assignments

For this question you may use Chernoff-Hoeffding bound, discussed in class and stated here for your convenience:

Additive Chernoff-Hoeffding Bound. For independent 0–1 random variables $X_1, \ldots, X_k$, write $S = \sum_i X_i$ and $\mu = \mathbb{E}S$. Then:

$$\Pr[S - \mu \geq a] \leq e^{-2a^2/k}, \qquad \Pr[S - \mu \leq -a] \leq e^{-2a^2/k} \quad (a \geq 0).$$

You may use the inequality $(1-u)^k \leq e^{-uk}$ for $0 \leq u \leq 1$

For a Boolean formula φ on v variables, let $p = \Pr_{z \sim \{0,1\}^v}[\varphi(z) = 1]$, where $z \sim \{0,1\}^v$ means that z is randomly chosen from the uniform distribution on $\{0,1\}^v$.

(a) Give a randomized algorithm that outputs $\widehat{p}$ with $\Pr[|\widehat{p} - p| > \varepsilon] \leq \delta$, using $O(\varepsilon^{-2} \ln(1/\delta))$ evaluations of φ. Here $0 < \varepsilon, \delta < 1/2$. Prove the guarantee.

(b) Explain why this does not give a randomized polynomial-time algorithm for SAT. Consider the difference between an unsatisfiable formula and a formula with exactly one satisfying assignment. What happens to your sampling bound when the additive error must be smaller than half that difference?