Homework 2, Computational Complexity 2025

The deadline is 23:59 on Monday 10 November. Please submit your solutions on Moodle. Typing your solutions using LaTeX is strongly encouraged. The problems are meant to be worked on in groups of 1–2 students. Please submit only one writeup per team. You are strongly encouraged to solve these problems by yourself. If you must, you may use books or online resources to help solve homework problems, but you must credit all such sources in your writeup and you must never copy material verbatim.

1 A *k-query oracle Turing machine* is an oracle Turing machine that is permitted to make at most k queries on each input. Define $\mathsf{P}^{A,k}$ to be the collection of languages that are decidable by polynomial-time k-query oracle Turing machines with an oracle for A. Note that

$$\mathsf{NP} \cup \mathsf{coNP} \subseteq \mathsf{P}^{\mathrm{SAT},1}.$$

Show that, if $\mathsf{NP} \neq \mathsf{coNP}$, then the above inclusion is strict, that is, $\mathsf{NP} \cup \mathsf{coNP} \neq \mathsf{P}^{\mathrm{SAT},1}$.

Solution: Consider the language $L \coloneqq \mathrm{SAT} \times \{1\} \cup \mathrm{UNSAT} \times \{0\}$. Observe that $L \in \mathsf{P}^{\mathrm{SAT},1}$, on input (ϕ, b) we query $\phi \in \mathrm{SAT}$ and accept if the oracle's answer coincides with b.

On the other hand if $L \in \mathsf{NP} \cup \mathsf{coNP}$, we can show that $\mathsf{NP} = \mathsf{coNP}$ and get a contradiction with the assumption. Observe that $\mathrm{SAT} \leq L$ and $\mathrm{UNSAT} \leq L$, the reductions are $\phi \mapsto (\phi, 1)$ and $\phi \mapsto (\phi, 0)$ respectively. Then if $L \in \mathsf{NP}$, then $\mathrm{UNSAT} \in \mathsf{NP}$, so $\mathsf{NP} = \mathsf{coNP}$. Analogously, if $L \in \mathsf{coNP}$, then $\mathrm{SAT} \in \mathsf{coNP}$, so $\mathsf{NP} = \mathsf{coNP}$.

2 Construct an oracle A such that $\mathsf{BPP}^A \not\subseteq \mathsf{NP}^A$. Namely, consider the class of oracles

$$\mathcal{A} \coloneqq \left\{ A \subseteq \{0,1\}^* : \forall n \geq 2, |A \cap \{0,1\}^n|/2^n \in \{\tfrac{1}{4}, \tfrac{3}{4}\} \right\}$$

and the associated language $L_A = \left\{1^n : |A \cap \{0,1\}^n|/2^n = \tfrac{3}{4}\right\}$. Show that

(i) $L_A \in \mathsf{BPP}^A$ for every $A \in \mathcal{A}$.
(ii) $L_A \notin \mathsf{NP}^A$ for some $A \in \mathcal{A}$.

Solution:

(i) Fix $A \in \mathcal{A}$ and consider the following PTM with oracle access to A. $\mathcal{M}$ on input $x \in \{0,1\}^n$:
   (a) if $x \neq 1^n$, **reject**
   (b) sample $y \in \{0,1\}^n$ uniformly at random
   (c) if $A(y) = 0$, **reject** else **accept**

Note that $\mathcal{M}$ rightly rejects any input which is not of the form 1^n. Suppose now that $1^n \in A$. Then $|A \cap \{0,1\}^n| = 3 \cdot 2^n/4$ and so that the probability that $A(y) = 1$ is $3/4$. On the other hand if $1^n \notin A$, then by the promise of $\mathcal{A}$, $|A \cap \{0,1\}| = 2^n/4$ and so the probability that $A(y) = 0$ is $3/4$. This shows that $\mathcal{M}$ has success probability $\geq 3/4$ over all inputs and $L_A \in \mathsf{BPP}^A$ for all $A \in \mathcal{A}$.

(ii) Let $(N_i)_{i\in\mathbb{N}}$ be an ordering of all non-deterministic Turing machines. We show how to build an $A \in \mathcal{A}$ that makes each NDTM err on some input. The construction of A will be iterative and to simplify notation we see A as a function $A : \{0,1\}^* \to \{0,1,*\}$ where $A(x) = 0$ means $x \notin A$, $A(x) = 1$ means $x \in A$ and $A(x) = *$ means that the membership is not yet decided. We initialize A with $*$ everywhere and will fix values for A on strings of increasing size. Suppose that we have dealt with all machines up to N_t and let $n \in \mathbb{N}$ be the smallest size of a string x with $A[x] = *$. We will ensure that N_t fails to decide whether $1^n \in L_A$ correctly. To do so, run N_t on 1^n with oracle A. Every time N_t queries A for some string $x \in \{0,1\}^p$:

- If $A[x] \neq *$, answer with $A[x]$
- If $A[x] = *$, it must be that $A(x) = *$ for each $x \in \{0,1\}^p$. Set A on strings of size p such that $|A \cap \{0,1\}^p| = (3/4) \cdot 2^n$.

If N_t reject 1^n, N_t already errs on 1^n because $1^n \in L_A$. If N_t accepts 1^n, there exists a non-deterministic branch that accepts with queries $Q = (x_1, \ldots, x_q)$. Because N_t is a NDTM, $q \leq n^k$ and without loss of generality, we may assume $n^k \leq 2^n/100$ (if not, start with a larger n). Since Q reads a tiny fraction of the 2^n entries of A at level n, we may switch $2^n/2$ 1-values into 0-values, effectively making $1^n \notin L_A$ while fooling Q and making N_t still accepting. This process produces an $A \in \mathcal{A}$ that fools each NDTM on some input and hence $L_A \notin \mathsf{NP}^A$.

Common mistakes in (ii):

- Observing that for some machine M we have $M^A(1^n) = 0$, then changing the values of $A \cap \{0,1\}^n$ such that the fraction of included values becomes $3/4$. If done directly, this may result in $M^{A'}(1^n) = 1$, since the set of queries M makes *across all computational paths* may include the entire $\{0,1\}^n$. This can be fixed by enforcing that $M^A(1^n) = 0$ regardless of the oracle answers, but then a more careful analysis of the complementary case is required. `[15 points]`

- Same as the previous one, but trying to avoid changing all potential oracle queries that are made across all computational paths. This has inherent issue with the fact that on input 1^n M^A can query *all* values in $\{0,1\}^n$, since it is allowed to run for $\mathsf{poly}(n) \gg n$ time. `[17 points]`

3 (Crazy variations of BPP)

3a Define a new class $\mathsf{BPP}_{\mathsf{path}}$ by changing the definition of BPP as follows. We have $L \in \mathsf{BPP}_{\mathsf{path}}$ if there is a polynomial-time probabilistic TM M such that if $x \in L$ then the fraction of all computation paths (leaves) of $M(x)$ that are accepting is $\geq 2/3$, and if $x \notin L$ then the fraction of accepting paths is $\leq 1/3$. Note that the computation tree of $M(x)$ need not be balanced! Show that

$$\mathsf{NP} \subseteq \mathsf{BPP}_{\mathsf{path}}.$$

Solution: Let us show that SAT $\in \mathsf{BPP}_{\mathsf{path}}$. On the input ϕ, where ϕ is a CNF over n-variables, let us toss n random coins to get a random string $r \in \{0,1\}^n$. If $\phi(r) = 1$ we toss additional $n+2$ coins and accept, if $\phi(r) = 0$ we reject without tossing any additional coins. Then is ϕ has k satisfying assignments, the number of accepting paths is $k2^{n+2}$, and the number of rejecting paths is $2^n - k$. If $\phi \notin$ SAT we have $k = 0$, so all paths are rejecting, and if $\phi \in$ SAT, then $k \geq 1$, so $k2^{n+2}/3 > 2^n \geq 2^n - k$, so $(2/3)k2^{n+2} \geq 2/3(k2^{n+2} + 2^n - k)$, hence the accepting paths are 2/3-fraction of all paths.

3b Define another class $\mathsf{PostBPP}$ as follows. We have $L \in \mathsf{PostBPP}$ if there is a polynomial-time probabilistic TM M that on each run outputs a symbol in $\{0, 1, \perp\}$ with the property that $\Pr[M(x) \neq \perp] > 0$ for all x, and also that $\Pr[M(x) = L(x) \mid M(x) \neq \perp] \geq 2/3$. That is, we have a PTM that can sometimes—but not always—output $\perp$ (for "abort"), but conditioned on it not aborting it gives the correct 0/1 answer with high probability. Show that

$$\mathsf{BPP}_{\mathsf{path}} = \mathsf{PostBPP}.$$

Solution: Observe that the solution for **3a** can be easily adapted to show that $\mathsf{PostBPP} \subseteq \mathsf{BPP}_{\mathsf{path}}$: suppose $L \in \mathsf{PostBPP}$, and M is the TM witnessing the inclusion. Let us run M 10 times and output $\perp$ if any of the runs returns $\perp$, otherwise output the majority of the outputs. Then we have by Chernoff bound that $\Pr[M'(x) = L(x) \mid M'(x) \neq \perp] \geq 0.9$, where M' is the result of the 10-fold repetition. Wlog assume that all computation paths in M' have the same length n^c for a fixed constant c.

Then define M'' as follows:

- Run $M'(x)$.
- If $M'(x) \in \{0,1\}$, toss $n^c + 2$ coins and return $M'(x)$.
- If $M'(x) = \perp$, reject.

Suppose that $\Pr[M'(x) \neq \perp] = p$, so there are $p2^{n^c}$ paths in M' that lead to 0/1 answer. Then if $x \notin L$, then there are at most $2^{n^c+2}0.1p2^{n^c}$ accepting paths and $2^{n^c+2}p2^{n^c} + (1-p)2^{n^c} \geq p2^{2n^c+2}$ paths in total, so the fraction of accepting paths is at most 0.1. If $x \in L$, then there are at least $0.9p2^{2n^c+2}$ accepting paths, so

$$\frac{0.9p2^{2n^c+2}}{p2^{2n^c+2} + (1-p)2^{n^c}} \geq \frac{0.9p2^{n^c+2}}{(p + 2^{-n^c}/4)2^{n^c+2}} \geq \frac{0.9p2^{n^c+2}}{(5/4p)2^{n^c+2}} = 0.9/1.25 > 2/3.$$

Now it remains to show that $\mathsf{BPP}_{\mathsf{path}} \subseteq \mathsf{PostBPP}$. Suppose that $L \in \mathsf{BPP}_{\mathsf{path}}$ and M is the TM witnessing this. Suppose that M runs in time n^c. Then let us modify M such that at any branch it makes exactly n^c coin tosses and the number of 0-branches and 1-branches in the new machine coincides with that for the old one. We define the new machine M' as follows:

- Run $M(x)$ and count the number of coin tosses it made.
- Suppose the number of coin tosses is m. Make $n^c - m$ additional coin tosses to get a string $r \in \{0,1\}^{n^c-m}$.
- If $r = 1^{n^c-m}$ return $M(x)$, otherwise return $\perp$.

Then M' is still polynomial-time probabilistic TM, it makes exactly n^c coin tosses in every run, and the number of 0-branches and 1-branches coincides with that of M. Moreover $M'(x) \in \{0,1\}$ implies $M'(x) = M(x)$. Therefore $\Pr[M'(x) = L(x) \mid M'(x) \neq \perp] \geq 2/3$, hence $L \in \mathsf{PostBPP}$.