Problem Set 3

- Due date: **15 Apr, 2023** (released on 29 Mar, 2023)
- The points for each problem is indicated on the side. The total for this set is **100** points but you are expected to answer any **80 points** worth of questions for a full score (anything additional would nevertheless be included in your aggregate score).
- The problem set has a fair number of questions so please do not wait until close to the deadline to start on them. Try and do one question every couple of days.
- Turn in your problem sets electronically (PDF; either LaTeXed or scanned etc.) via email.
- Collaboration with other students taking this course is encouraged, but collaboration with others is not allowed. Irrespective of this, all writeups must be done individually and must include names of all collaborators (if any).
- Referring to sources other than the text book and class notes is STRONGLY DISCOURAGED. But if you do use an external source (eg.,other text books, lecture notes, or any material available online), ACKNOWLEDGE all your sources (including collaborators) in your writeup. This will not affect your grades. However, not acknowledging will be treated as a serious case of academic dishonesty.
- Be clear in your writing.

---

1. **[Problems in** PSPACE**]** (10 + 10)

   (a) Antakshari: Let Σ be a finite alphabet and S be a finite set of strings in Σ^*. For a string $s_1 s_2 \dots s_n$ (with each $s_i \in \Sigma$), let $\text{last}(s) = s_n$ and $\text{first}(s) = s_1$. Let P_0, P_1 be two players playing this game, starting with P_0 and turns alternating between the two players. In the first turn, P_0 can choose any $s \in S$. For all subsequent turns, the other player is to choose a string $s' \in S$ such that s' has not been chosen before and that $\text{last}(s) = \text{first}(s')$. The first player who is unable to choose the next string loses. Define the following language based on the above game:

   $$\text{Antakshari} = \left\{ \langle S \rangle \ : \ \begin{array}{c} S \text{ is a finite set of strings for which the} \\ \text{starting player } P_0 \text{ has a winning strategy} \end{array} \right\}$$

   Show that Antakshari $\in$ PSPACE.

   (For extra credit, show that it is actually PSPACE-complete)

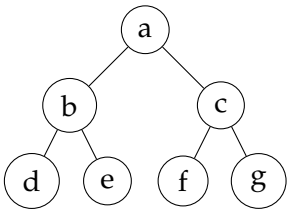
   (b) (This problem requires some prior exposure with automata and regular expressions. If you are unfamiliar with them, please get in touch with me and I will tell you what you need to know for this.)

   Let $\text{EQ}_{\text{RegExp}} = \{\langle R, S \rangle \ : \ R, S \text{ are equivalent regular expressions}\}$.
   Show that $\text{EQ}_{\text{RegExp}} \in$ PSPACE. (For extra credit, show it is PSPACE-complete.)

   [Hint: It might actually be easier to show that $\text{EQ}_{\text{RegExp}} \in \text{coNPSPACE}$]

2. **[Boolean Formula Evaluation]** (7 + 8)

   (a) Prove that computing the DFS order (the order of vertices visited, including repetitions, in a DFS traversal that starts at the root and ends at the root) of an *undirected binary tree* $T = (V, E)$ can be done in L (logspace).
   For example, the DFS order of the tree below is $a, b, d, b, e, b, a, c, f, c, g, c, a$.



   [Hint: (a) You may assume that the tree is described as follows: For every vertex $v \in V$, there is a function $next_v : V \to V \cup \{\top\}$ which gives a clockwise ordering of the edges around the vertex V. I.e., For every vertex v, there is a cyclic ordering among the neighbours of v and $next_v(n)$ is the next neighbour in this cyclic ordering if n is a neighbour of v and $\top$ otherwise. Finally, check that one can make this assumption without loss of generality.]

   (b) A Boolean formula φ on n inputs is a directed tree with n *sources* (vertices with no incoming edges) and one *sink* (vertex with no outgoing edges). All nonsource vertices are called *gates* and are labeled with one of $\vee, \wedge$ or $\neg$. The vertices labeled with $\vee$ or $\wedge$ have fan-in 2 and the vertices labeled with $\neg$ have fan-in 1. Let $x \in \{0,1\}^n$ be some input. The output of φ on x, denoted by $\varphi(x)$, is defined in the natural way. The Boolean formula evaluation problem deals with, given a formula φ on n inputs and $x \in \{0,1\}^n$, computing the value of $\varphi(x)$. Show that formula evaluation can be done in logspace. More precisely, define

$$\text{FormulaEval} = \{\langle \varphi, x \rangle \ : \ \varphi \text{ is a Boolean formula and } \varphi(x) = 1\}$$

   Prove that FormulaEval $\in$ L.

3. **[Generalising Karp-Lipton-Sipser]** (2 + 3 + 2 + 3 + 5)

   (i) Consider the following language L corresponding to the encodings of *True* expressions of the form

$$\text{“}(\exists x \in \{0,1\}^m \ \varphi(x)) \Leftrightarrow (\forall y \in \{0,1\}^m \ \psi(y))\text{”}$$

   where φ and ψ are some polynomial time computable predicates.
   Show that $L \in$ PH. At what level of the hierarchy is it in?

   (ii) Let $L \in$ NP be any language that you are promised is in P/poly. This means that there is a sequence of “advice strings” $\{z_i\}_{i=1}^{\infty}$ and a polynomial time deterministic TM M such that for all x we have $x \in L \Leftrightarrow M(x, z_{|x|}) = 1$.
   Define the following language:

$$\text{ValidAdvice}_L = \left\{(z, n) \ : \ \forall x \in \{0,1\}^n \ x \in L \Leftrightarrow M(x, z) = 1\right\}$$

   Show that ValidAdvice$_L \in$ PH. What level of the hierarchy is it in?

(iii) Try and give a different proof of the Karp-Lipton-Sipser theorem using the above observations.

(iv) Suppose $L \in \mathsf{coNP}$ that is promised to be in $\mathsf{NP/poly}$. Show that $\text{ValidAdvice}_L \in \mathsf{PH}$. What level of the hierarchy is it in?

(v) Show that if $\mathsf{coNP} \subseteq \mathsf{NP/poly}$, then PH collapses. (To what level?)

4. **[NEXP and coNEXP with advice]** (10)

Prove that $\mathsf{coNEXP} \subseteq \mathsf{NEXP/poly}$.

(This is in contrast to the previous problem where you show that it is unlikely that $\mathsf{coNP} \subseteq \mathsf{NP/poly}$)

[Hint: Recall the "advice" used the proof of the Immerman-Szelepcsényi theorem to help an NL machine realise there *no* path of legnth i from s to t. Is there a similar advice that you can give to an NEXP machine to simulate a coNEXP machine on any input of a certain length?]

5. **[Kannan's theorem]** (7 + 8 + 5)

(i) Fix any constant $c > 0$. Show that there is a language $L \in \mathsf{PH}$ that is not in $\text{SIZE}(n^c)$.

[Hint: Can you try and encode "the lexicographically smallest circuit of size $10n^c$ that is not computable by c ircuits of size n^c" as a quantified expression?]

(ii) Show that, for any constant $c > 0$, there is a language in $L \in \Sigma_2^p$ that is not in $\text{SIZE}(n^c)$.

[Hint: Either $\mathsf{NP} \subseteq \mathsf{P/poly}$ or not...]

(iii) Note that this means in particular that, for any constant $c > 0$, we know NP is not computable by circuits of size n^c. Why does this not show that $\mathsf{NP} \not\subseteq \mathsf{P/poly}$ (which, if you recall, is stronger than saying $\mathsf{P} \neq \mathsf{NP}$)?

6. **[VC Dimension (Problem 5.13)]** (3 + 7 + 10)

The Vapnik-Chervonenkis dimension (VC-dimension) is a very important concept in learning theory. Let $\mathcal{S} = \{S_1, \ldots, S_m\}$ be a collection of subsets of a finite universe U. We shall say that a set $X \subseteq U$ is *shattered* by $\mathcal{S}$ if, for every $X' \subseteq X$, there is some $i \in [m]$ such that $X' = X \cap S_i$.

The VC-dimension of a collection $\mathcal{S}$ is defined as the largest k such that there is some set $X \subseteq U$ of size k that is shattered by $\mathcal{S}$.

Consider the following succinct version of providing the family $\mathcal{S}$ via a Boolean circuit C with $2n$ input bits (which will be encoding a collection $\mathcal{S}$ of 2^n subsets of the universe $U = \{0,1\}^n$):

$$S_i = \left\{x \in \{0,1\}^n \ : \ C(i,x) = 1\right\}$$

(where i is provided to the circuit in binary, hence requiring only n bits).

Define the language Circuit-VC-Dim as

$$\text{Circuit-VC-Dim} = \{(C,k) \ : \ C \text{ encodes (as above) a collection of VC-dimension} \geq k\}$$

(i) Show that if $\mathcal{S}$ is the collection encoded (as above) by a Boolean circuit C with $2n$ input bits, then VC-dimension$(\mathcal{S}) \leq n$.

(ii) Show that Circuit-VC-Dim $\in \Sigma_3^P$.

(iii) Show that Circuit-VC-Dim is Σ_3^P complete.

]

[Hint: It would be convenient to reduce from Σ_3-SAT. Consider an instance $\exists p \forall q \exists r \ \varphi(p,q,r)$ where each of p,q,r are n-bit strings. Let $U = \{0,1\}^n \times [n]$. Try and build a collection $\mathcal{S} = \{S_{pqr} \ : \ p,q,r \in \{0,1\}^n\}$ with $S_{pqr} \subseteq \{(p,1),\ldots,(p,n)\}$ that can be encoded by a small circuit. (Observe that the VC-dimension of such a collection cannot be more than n as each set is of size at most n.)

Build this collection so that you can prove $\exists p \forall q \exists r \ \varphi(p,q,r)$ is true if and only if some set of size n (in fact, a set of the form $X = \{(p,1),\ldots,(p,n)\}$) is shattered by this collection. That is,

$$\text{VC-dimension}(\mathcal{S}) \geq n \iff \exists p \forall q \exists r \ \varphi(p,q,r).$$