Some policies: (See the course policy at http://madhu.seas.harvard.edu/courses/Fall2020/policy.html for the full policies.)

- **Collaboration:** You can collaborate with other students that are currently enrolled in this course (or, in the case of homework zero, planning to enroll in this course) in brainstorming and thinking through approaches to solutions but you should write the solutions on your own and cannot share them with other students.

- **Owning your solution:** Always make sure that you “own” your solutions to this other problem sets. That is, you should always first grapple with the problems on your own, and even if you participate in brainstorming sessions, make sure that you completely understand the ideas and details underlying the solution. This is in your interest as it ensures you have a solid understanding of the course material, and will help in the midterms and final. Getting 80% of the problem set questions right on your own will be much better to both your understanding than getting 100% of the questions through gathering hints from others without true understanding.

- **Serious violations:** Sharing questions or solutions with anyone outside this course, including posting on outside websites, is a violation of the honor code policy. Collaborating with anyone except students currently taking this course or using material from past years from this or other courses is a violation of the honor code policy.

- **Submission Format:** The submitted PDF should be typed and in the same format and pagination as ours. Please include the text of the problems and write Solution X: before your solution. Please mark in gradescope the pages where the solution to each question appears. Points will be deducted if you submit in a different format.

- **Late Day Policy:** To give students some flexibility to manage your schedule, you are allowed a net total of eight late days through the semester, but you may not take more than two late days on any single problem set. No exceptions to this policy.

By writing my name here I affirm that I am aware of all policies and abided by them while working on this problem set:

Your name: (Write name and HUID here)

Collaborators: (List here names of anyone you discussed problems or ideas for solutions with)

No. of late days used on previous psets (not including Homework Zero):
No. of late days used after including this pset:
Questions

Please solve the following problems. Some of these might be harder than the others, so don’t despair if they require more time to think or you can’t do them all. Just do your best. Also, you should only attempt the bonus questions if you have the time to do so. If you don’t have a proof for a certain statement, be upfront about it. You can always explain clearly what you are able to prove and the point at which you were stuck. You can always write “I don’t know” and you will get 15 percent of the credit for this problem. If you are stuck on this problem set, you can use Piazza to send a private message to all staff.

Problem 1: Recall that in section, we defined the function $LOOKUP_k : \{0, 1\}^{2^k + k} \rightarrow \{0, 1\}$ by

$$LOOKUP_k(x[0], x[1], \ldots, x[2^k - 1], i[0], \ldots, i[k - 1]) = x[i[0] + 2i[1] + \cdots + 2^{k-1}i[k - 1]]$$

and proved that there exists a circuit with $O(2^k)$ gates that computes $LOOKUP_k$. We’ll write $LOOKUP_k(x, i)$ as shorthand for $LOOKUP_k(x[0], x[1], \ldots, x[2^k - 1], i[0], \ldots, i[k - 1])$.

Problem 1.1 (8 points) Prove that $LOOKUP_k(x, i)$ cannot be computed by circuits $C$ of size $o(2^k)$.

Solution 1.1:

Problem 1.2 (4 points) Prove that for every $x \in \{0, 1\}^{2^k}$ there exists a circuit $C(x)_1$ of size $o(2^k)$ that computes $LOOKUP_k(x, i)$. (You may use results stated but not proved in lecture.)

Solution 1.2:

Problem 1.3 (8 points) Prove that there exists $x \in \{0, 1\}^{2^k}$ such that there does not exist a circuit $C(x)_1$ of size $o(2^{\sqrt{k}})$ that computes $LOOKUP_k(x, i)$.

Solution 1.3:

Problem 2: Recall that

$$SIZE_{n,m}(s) = \{ f : \{0, 1\}^n \rightarrow \{0, 1\}^m \mid \exists \text{ NAND circuit } C \text{ of at most } s \text{ gates s.t. } C \text{ computes } f \}.$$ 

Let $BSIZE(s) = \cup_{n \in \mathbb{N}} SIZE_{n,1}(s)$ represent the set of Boolean functions (i.e., those whose output is in $\{0, 1\}$) that are computable by NAND circuits of size at most $s$.

Prove the following:

Problem 2.1 (3 points): $|BSIZE(s)| = 2^{O(s \log s)}$.

Solution 2.1:

Problem 2.2 (15 points): $|BSIZE(s)| = 2^{\Omega(s)}$.

Solution 2.2:

Problem 2.3 (2 points): $BSIZE(s) \subsetneq BSIZE(s^2)$, for every sufficiently large $s$.

Solution 2.3:

$^1$That is, the circuit may depend on $x$. 


Problem 3.1 (15 points): Describe the set of inputs accepted by the following DFA with 4 states given by \((T, S)\) where \(S = \{1, 3\}\) and \(T\) is given by the following table:

<table>
<thead>
<tr>
<th>Input ↓ \ State →</th>
<th>0</th>
<th>1</th>
<th>2</th>
<th>3</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>2</td>
<td>2</td>
<td>3</td>
<td>3</td>
</tr>
</tbody>
</table>

Solution 3.1:

Problem 3.2 (bonus, 0 points): For all \(n\), find, with proof, a regular expression with \(10n + 100\) characters that’s equivalent to some DFA with \(2^n\) states and not equivalent to any DFA with \(2^n - 1\) states. (Hint: generalize 3.1.)

Solution 3.2:

Problem 4 (15 points): Build a DFA that accepts the regular expression: \((aaa(a|b)^*)|((a|b)^*aba(a|b)^*)\).

Solution 4:

Problem 5 (20 points): Given a function \(f : \{0, 1\}^* \rightarrow \{0, 1\}\), let \(f^R : \{0, 1\}^* \rightarrow \{0, 1\}\) be the function given by \(f^R(x_1, \ldots, x_n) = 1\) if and only if \(f(x_n, \ldots, x_1) = 1\). Prove that if \(f\) is computed by a DFA, then so is \(f^R\).

Solution 5:

Problem 6 (20 points): Prove that the function \(f : \{0, 1\}^* \rightarrow \{0, 1\}\) given by \(f(x) = 1\) if and only if \(x \in \{0^m1^n10^m| m, n \in \mathbb{N}\}\) is not computed by any DFA.

Solution 6: