# Comp 330 (Fall 2024): Assignment 4

Answers must be submitted online by December 4 (11:59 pm), 2024.

## General instructions

- **Important:** All of the work you submit must be done by only you, and your work must not be submitted by someone else. Plagiarism is academic fraud and is taken very seriously.
- To some extent, collaborations are allowed. These collaborations should not go as far as sharing code or giving away the answer. **You must indicate on your assignments the names of the people with whom you collaborated or discussed your assignments (including members of the course staff). If you did not collaborate with anyone, write "No collaborators". If asked, you should be able to orally explain your solution to a member of the course staff.**
- It is your responsibility to guarantee that your assignment is submitted on time. We do not cover technical issues or unexpected difficulties you may encounter. Last minute submissions are at your own risk.
- Multiple submissions are allowed before the deadline. We will only grade the last submitted file. Therefore, we encourage you to submit as early as possible a preliminary version of your solution to avoid any last minute issue.
- Late submissions can be submitted for 24 hours after the deadline, and will receive a flat penalty of 20%. We will not accept any submission more than 24 hours after the deadline. The submission site will be closed, and there will be no exceptions, except medical.
- In exceptional circumstances, we can grant a small extension of the deadline (e.g. 24h) for medical reasons only. However, such request must be submitted before the deadline, justified and approved by the instructors.
- Violation of any of the rules above may result in penalties or even absence of grading. If anything is unclear, it is up to you to clarify it by asking either directly the course staff during office hours, by email at or on the discussion board on Ed. Please, note that we reserve the right to make specific/targeted announcements affecting/extending these rules in class and/or on the website. It is your responsibility to monitor Ed for announcements.
- The course staff will answer questions about the assignment during office hours or in the online forum. We urge you to ask your questions as early as possible. We cannot guarantee that questions asked less than 24h before the submission deadline will be answered in time. In particular, we will not answer individual emails about the assignment that are sent the day of the deadline.
- Unless specified, **you must show your work and all answers must be justified.**

1. (10 points) Let $M$ be a Turing machine and $w$ a word. Suppose that $M$ accepts $w$ without ever moving left. Show that $M$ accepts a word $w'$ such that, during the computation of the machine, the machine never moves left and never finds itself twice in the same state $(q, \delta)$ where $\sigma$ is the state of the machine and $\sigma$ the symbol under the tape head.

> **Solution:** Suppose that the property is not satisfied by $w$: then the machine has made two transitions $(q, \sigma) \mapsto (q', \sigma', D)$ with the same pair $(q, \sigma)$. If this situation occurs on the word $w$ at position $i$ and at position $j$, then we can remove from $w$ the subword $w[i \cdots j[$ and the machine will have exactly the same behavior on this new word since, not being able to go to the left, it cannot differentiate between these two entries. The word $w' = w[1 \cdots i[w[j \cdots [$ obtained is strictly shorter than $w$. By repeating this process a finite number of times, we produce a word $w'$ which therefore has the desired property.

2. Determine if the following properties are decidable. Justify your answer using the Rice's theorem if the property is undecidable or propose an algorithm to decide otherwise.

   (a) (10 points) Let $\mathcal{L}$ be a language over the alphabet $\Sigma = \{a, b\}$ such that the words $w \in \mathcal{L}$ have the same number of $a$ and $b$. Is the following property is decidable or not? "Determine if a Turing machine recognizes at least one word of $\mathcal{L}$".

   > **Solution:** We apply the Rice's theorem to show it is undecidable. The property is non-trivial. The language $\{a\}$ does not satisfy the property, while $\{ab\}$ does.

   (b) (10 points) Is the following property is decidable or not? "Determine whether a Turing machine M accepts at least one word without moving left" (Hint: You can use the results from the first question).

   > **Solution:** Rice's theorem does not apply because it is a property of a machine, not a language. From the previous question, if $M$ accepts a word without moving left, then it accepts a word without moving left and without ever repeating the (state,symbol) pair.
   > We see that such a word is accepted in less than $|Q| \cdot |\Sigma|$ steps, which is also a bound on its length. We can therefore enumerate all these words and simulate $M$ during this finite time to see if there is such a word.

3. Recall the Post Correspondence Problem (PCP) seen in class.

   (a) Solve (manually) the following PCP instances (i.e., give an explicit solution if there is one, or justify why it cannot be solved).

   i. (10 points) $P1 = \{(a, ab), (ba, aba), (b, aba), (bba, b)\}$

   > **Solution:** $(a, ab), (bba, b), (ba, aba)$ is a solution

   ii. (10 points) $P2 = \{(ab, bb), (aa, ba), (ab, abb), (bb, bab)\}$

**Solution:** It is impossible because the pairs 3 and 4 are making the word on the right growing faster than the one on the left. There is no way to restore the size. In addition, you cannot start a match with 1 or 2.

(b) (10 points) We call "PCP without repeat" (PCPWR) a variant of PCP in which a solution can use at most once each pair of the PCP instance. Show that PCPWR is decidable.

**Solution:** There is a finite number of sequence of indices possible. We just need to enumerate them all.

(c) (15 points) Using the reduction from universal Turing machines described in class, argue that PCPWR is $\mathcal{NP}$-hard (a sketch of a proof is sufficient). Then, conclude it is $\mathcal{NP}$-complete.

**Solution:** Consider a $\mathcal{NP}$-complete problem, which is decided by a non-deterministic Turing machine $M$ that runs in time $p(n)$. We use the set of pairs defined in the reduction presented in class. The number of pairs constructed is a function of the size of the alphabet and the number of states of the Turing machine $M$. The only pair you can start with is $(\#, \#q_0w\#)$.
The set of pairs needed to simulate $p(n)$ steps of $M$ is $f(p(n))$ where $f(n)$ is a polynomial function that expresses the number of dominoes to simulate $n$ steps of $M$. Thus, we accept $w$ by $M$ if and only if we have a positive instance of PCPWR. (Note: the constructed set could repeat certain pairs several times, but we restrict ourselves to use at most one copy of each of these pairs).
Since a solution can be verified in polynomial time, it is $\mathcal{NP}$-complete.

4. Given a graph $G = (V, E)$ and a pair of vertices $(u, v) \in V^2$, an Hamiltonian path is a path from $u$ to $v$ that visits each vertex exactly once. The Hamiltonian cycle (decision) problem is to determine if $G$ has an Hamiltonian cycle, and the Hamiltonian path (decision) problem is to determine if it exists a Hamiltonian path from $u$ to $v$ in $G$.
We will show that the Hamiltonian path problem is $\mathcal{NP}$-complete. Here, you assume that the Hamiltonian cycle problem is $\mathcal{NP}$-complete (See textbook for a full proof).

(a) (10 points) Show that the Hamiltonian path problem is in $\mathcal{NP}$.

**Solution:** The Hamiltonian path problem is in $\mathcal{NP}$ because given a path, we can check in polynomial time whether it passes once and only once through each vertex of the graph, and whether it has $u$ and $v$ as its endpoint.

(b) (15 points) Then, we show that the Hamiltonian path problem is $\mathcal{NP} - complete$. To achieve this, we will build a polynomial-time reduction from the Hamiltonian Cycle problem. Let $\mathcal{I} = \langle G = (V, E)\rangle$ be an instance of the Hamiltonian Cycle problem. We will transform the instance $\mathcal{I}$ into an instance $\mathcal{I}'$ of the Hamiltonian path problem. More specifically, we will construct a graph $G' = (V', E')$ such that:

- Let $u$ be an arbitrary vertex of $V$,
- Let $v$ is a vertex not belonging to $V$ and set $V' = V \cup \{v\}$.

- $E' := E \cup \{(v, l) \mid l$ is a neighbor of $u$ in $G\}$.

Continue the proof...

**Solution:** This transformation can be done in polynomial time (i.e., we copy the graph G by adding a vertex and edges). We can prove that:

- If there exists a Hamiltonian cycle in $G$, then there exists a Hamiltonian path in $G'$. Let $C = (u, l_1, \cdots, l_{n-1}, u)$ be a Hamiltonian cycle in $G$. Then, build the path $P = (u, l_1, \cdots, l_{n-1}, v)$ in $G'$. This path is Hamiltonian as it passes once and only once through $v$ and through each vertex of $G$ since $C$ is a Hamiltonian cycle.
- If there exists a Hamiltonian path in $G'$, then there exists a Hamiltonian cycle in $G$. Let $P = (u, l_1, \cdots, l_{n-1}, v)$ be a path in a graph $G'$. The cycle $C = (u, l_1, \cdots, l_{n-1}, u)$ is Hamiltonian for the graph $G$.

Therefore, we reduced the Hamiltonian path to the Hamiltonian cycle, and the Hamiltonian path problem is $\mathcal{NP}$-complete.

5. **[For extra credit.]** This is a question about the learning automata tutorial videos. To complete this question, you may want to consult the tutorial videos under the Tutorial 6 tab on myCourses.

   You are given an unknown regular language $L \subseteq \{a, b\}^*$ for which you would like to learn a DFA $M$. To do so, you will run the $L^*$ algorithm presented in Tutorial 6 in which you are given a teacher $T$ which can answer membership questions and verify candidate DFA. We have broken down the execution of the $L^*$ algorithm into multiple sub-parts. Provide answers (i.e., traces of the algorithm run) for each of the sub-parts.

   (a) (5 points (bonus)) Suppose that the teacher $T$ initially provides you with the following membership answers: $T(\varepsilon) = 0, T(a) = 0, T(b) = 0$. Construct the initial observation table $O_0$. Show that $O_0$ is closed and consistent and build the corresponding DFA $M_{O_0}$.

   (b) (5 points (bonus)) You provide $M_{O_0}$ to $T$ as a candidate DFA. $T$ answers "No" and provides you with a counter-example: $T(bb) = 1$. You are also given the additional following membership answers: $T(ba) = 0, T(bba) = 1, T(bbb) = 0$. Using this information, build the new observation table $O_1$. Show that $O_1$ is closed, but not consistent.

   (c) (5 points (bonus)) Fix the inconsistency in $O_1$ by leveraging the following additional information: $T(ab) = 0, T(bab) = 1, T(bbab) = 0, T(bbbb) = 0$. Show that the new table $O_2$ is closed and consistent. Build the corresponding DFA $M_{O_2}$.

   (d) (5 points (bonus)) You provide $M_{O_2}$ to $T$. $T$ answers "No" and provides you with a counter-example: $T(b^5) = 0$. You are also given the additional following membership answers (represented as a piecewise function for compactness):

   $$T(b^n \cdot x) = \begin{cases} 1 & \text{if } n = 2 \ \& \ x \in \{\varepsilon, a\} \\ 0 & \text{if } 3 \leq n \leq 6 \ \& \ x \in \{\varepsilon, a, b, ab\} \end{cases}$$

   Using this information, build the new observation table $O_3$. Show that $O_3$ is closed, but not consistent.

   (e) (5 points (bonus)) Fix the inconsistency in $O_3$ by leveraging the following additional information (again, represented as a piecewise function for compactness)

   $$T(abb) = 1 \ \& \ T(b^n \cdot x) = \begin{cases} 0 & \text{if } n = 8 \ \& \ x = \varepsilon \\ 0 & \text{if } 1 \leq n \leq 5 \ \& \ x = abb \end{cases}$$

   Show that the new table $O_4$ is closed and consistent and build its corresponding DFA $M_{O_4}$.

   (f) (5 points (bonus)) You provide $M_{O_4}$ to $T$. $T$ answers "Yes!". What was the unknown language $L$? Provide a one sentence natural language description of $L$.

| Question: | 1 | 2 | 3 | 4 | 5 | Total |
|---|---|---|---|---|---|---|
| Points: | 10 | 20 | 45 | 25 | 0 | 100 |
| Bonus Points: | 0 | 0 | 0 | 0 | 30 | 30 |
| Score: | | | | | | |