Exercise 1: The class $\mathcal{P}$

1. Which of the following statements about the class $\mathcal{P}$ are true?
   
   - $\mathcal{P}$ is the class of all languages that are decidable by deterministic multi-tape Turing machines running in polynomial time.
   - A language $L$ belongs to $\mathcal{P}$ iff there is a constant $k$ and a decider $M$ running in time $O(n^k)$ such that $L = L(M)$.
   - A language $L$ belongs to $\mathcal{P}$ iff $L$ is decided by an $O(2^n)$ time DTM.
   - $A_{TM}$ belongs to $\mathcal{P}$.

2. Show that the following language (≥ decision problem) $18$-$\text{DOMINATINGSET} := \{\langle G \rangle \mid G \text{ has a dominating set of size at most 18} \}$ is in the class $\mathcal{P}$.

   Remark: A subset of the nodes of a graph $G$ is a dominating set if every other node of $G$ is adjacent to some node in the subset.

   Is $18$-$\text{DOMINATINGSET}$ decidable?

Sample Solution

1. True. Note that any multi-tape TM running in polynomial can be simulated by a single-tape TM running also in polynomial time (it is only quadratically slower).
   - True.
   - False. The implication from left to right of course holds, but the one from right to left does not hold.
   - False. The language $A_{TM}$ does not belong to $\mathcal{P}$ because $A_{TM}$ is an undecidable language and $\mathcal{P}$ contains only decidable languages.

2. A dominating set of size 18 exists if and only if all nodes are covered by a subset of the nodes $D \subseteq V$ with $|D| = 18$. There are less than $n^{18}$ sets with $|D| = 18$. We iterate through all of the sets (e.g., by using 18 nested for loops iterating over the identifiers of the node set if we number the nodes from 1 to $n$). Then for each set we test whether it dominates the whole graph by testing whether every node not in the set has a neighbor in the set. If this is the case for some set we accept. We reject if it’s not the case for all sets. This test can be done in time $O(n^2)$ for each other node by scanning the edge set. There are at most $|V \setminus D| \leq n$ nodes outside of $D$ for a $D$ with $|D| = 18$. So to check whether a specific $D$ is a dominating set we only need $O(n^3)$ time. In total the time complexity is $O(n^{18} \cdot n^3) = O(n^{21})$. Therefore $18$-$\text{DOMINATINGSET} \in \mathcal{P}$.
   - Yes, since $18$-$\text{DOMINATINGSET} \in \mathcal{P}$ and $\mathcal{P}$ contains only decidable languages.
Exercise 2: The class \( \mathcal{NP} \) and \( \mathcal{NPC} \)

1. Is the following true: A language \( L \) is decidable by an \( O(\log n) \) time deterministic single tape TM, then \( L \) belongs to \( \mathcal{NP} \).

2. Given a set \( U \) of \( n \) elements (’universe’) and a collection \( S \subseteq \mathcal{P}(U) \) of subsets of \( U \), a selection \( C_1, \ldots, C_k \in S \) of \( k \) sets is called a set cover of \( (U, S) \) of size \( k \) if \( C_1 \cup \ldots \cup C_k = U \).

Show that the problem

\[
\text{SetCover} := \{ \langle U, S, k \rangle \mid U \text{ is a set, } S \subseteq \mathcal{P}(U) \text{ and there is a set cover of } (U, S) \text{ of size } k \}
\]

is NP-complete.

You may use that

\[
\text{DominatingSet} = \{ \langle G, k \rangle \mid G \text{ has a dominating set with } k \text{ nodes} \}.
\]

is NP-complete.

3. Why can’t we solve \( \text{DominatingSet} \) in polynomial time the same way we solve \( 18-\text{DominatingSet} \)?

Sample Solution

1. True. Since \( O(\log n) \subseteq O(n^k) \) for any constant \( k \geq 1 \), language \( L \) can be solved in polynomial time by a deterministic single tape TM, hence \( L \) is in class \( \mathcal{P} \). Also, \( \mathcal{P} \subseteq \mathcal{NP} \), thus \( L \) is in \( \mathcal{NP} \).

2. SetCover is in NP: Guess a collection \( C_1, \ldots, C_k \in S \) of \( k \) sets from \( S \). Go through all elements of \( U \) and check if it is in one of the \( C_i \). This takes polynomial time.

SetCover is NP-hard: We reduce \( \text{DominatingSet} \) to SetCover. Let \( G = (V, E) \) be a graph and \( k \) an integer. We define a SetCover instance in the following way: We choose \( V \) to be the universe, i.e., \( U = V \) and \( S := \{ \Gamma_G(v) \mid v \in V \} \) where \( \Gamma_G(v) \) consists of the vertex \( v \) and all vertices adjacent to \( v \) in \( G \). This conversion takes polynomial time. Then \( \Gamma_G(v_1), \ldots, \Gamma_G(v_k) \) is a set cover of \( (U, S) \) iff \( v_1, \ldots, v_k \) is a dominating set of \( G \). Hence, \( \langle U, S, k \rangle \in \text{SetCover} \) iff \( \langle G, k \rangle \in \text{DominatingSet} \).

3. To prove \( \text{DominatingSet} \) in \( \mathcal{P} \), we would need to find a constant \( c \), and an associated \( O(n^c) \) algorithm, which would decide on ”any” instance \( (G, k) \), whatever \( k \) is. However, if we use the same brute force algorithm in exercise 1 to solve \( \text{DominatingSet} \), once we take the instance \( (G, k) \) to be checked, then it will run on \( O(n^c) \) for some constant \( c \), but here we are choosing \( c \) after we have seen \( k \), for arbitrary \( k \). So as \( k \) increases it approaches to \( n/2 \), then using the same brute force algorithm will yield in an exponential of order \( n/2 \) time complexity, and this does not prove that \( \text{DominatingSet} \in \mathcal{P} \).

Exercise 3: Regular and Context Free Languages

1. Is the language \( L := \{ w \in \text{DominatingSet} \mid |w| \leq 2021 \} \) regular? Is it decidable?

2. Using the pumping lemma for regular languages, show that the following language \( L_1 = \{ 0^m \mid m \text{ is a prime} \} \) is not regular.

3. Using the Pumping Lemma for CFL, show also that \( L_1 \) is not a CFL.
Sample Solution

1. We know that all words in the language have length at most 2021, hence $L$ can only contain a finite number of strings, thus $L$ is a finite language. And finite languages are regular since they can be easily described by a DFA or regular expressions, hence decidable.

2. $L_1$ can be proven, as follows, non regular using the pumping lemma for regular languages. Let $p$ be the pumping length. Let $t > p$ be a prime. Then, let $x = 0^i$, $y = 0^j$ and $z = 0^k$ such that $x + y + z = t$. Based on Pumping Lemma, for all $\ell \geq 0$, $xy^\ell z$ must also be in $L_2$. However, for $\ell = t + 1$, $xy^{t+1}z = 0^i0^j(t+1)0^k = 0^{(j+i+k)(t+1)} = 0^{(j+1)}$, where both $t$ and $j + 1$ are greater than 1. Therefore, $t(j + 1)$ is not a prime, and hence $xy^{t+1}z$ is not in $L_2$.

3. To prove that $L_2$ is not a context free language, let $p$ be the pumping length by considering the Pumping Lemma for context free languages. Let $t > p$ be a prime. Since $a^t$ is in $L_2$, let $uxvy^t z = 0^i0^j10^k = 0^{(j+i+k)}0^t = 0^{(j+1)}$, where both $t$ and $j + 1$ are greater than 1. Note that $|vy| > 0$ and $|vxy| \leq p$. Hence, $i + j > 0$ It must hold that $uv^{t+1}xy^{t+1}z \in L_2$. However, $uv^{t+1}xy^{t+1}z = uvxy^t z \cdot v^t y^t = 0^{(1+i+j)}$. Since both $t$ and $1 + i + j$ are greater than 1, $t(1 + i + j)$ is not a prime. Hence, $uv^{t+1}xy^{t+1}z$ is not in $L_2$. 
