University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
Z. Parsaeian

Theoretical Computer Science - Bridging Course
Sample Solution Exercise Sheet 8

Due: Monday, 11th of December 2023, 12:00 pm

Exercise 1: Class $\mathcal{P}$ *(1+3+3 Points)*

$\mathcal{P}$ is the set of languages ($\cong$ decision problems) which can be decided by an algorithm whose runtime can be bounded by $p(n)$, where p is a polynomial and n the size of the respective input (problem instance). Show that the following languages are in the class $\mathcal{P}$. Since it is typically easy (i.e. feasible in polynomial time) to decide whether an input is well-formed, your algorithm only needs to consider well-formed inputs. Use the $\mathcal{O}$-notation to bound the run-time of your algorithm.

(a) $\textsc{Palindrome} := \{w \in \{0,1\}^* \mid w \text{ is a Palindrome}\}$

(b) $4\text{-}\textsc{Clique} := \{\langle G\rangle \mid G \text{ has a } \textit{clique} \text{ of size at least } 4\}$

(c) $5\text{-}\textsc{VertexCover} := \{\langle G\rangle \mid G \text{ has a } \textit{vertex cover} \text{ of size at most } 5\}$.

Remarks:

- In both problems G is an undirected, simple graph.
- A *clique* in a graph $G = (V, E)$ is a set $C \subseteq V$ such that for all $u, v \in C : \{u, v\} \in E$.
- A *vertex cover* of $G = (V, E)$ is a subset $C \subseteq V$ of nodes, such that for all $\{u, v\} \in E$ it holds that $u \in C$ or $v \in C$.

Sample Solution

(a) We have already seen in sheet 5 that the problem can be solved with a Turing machine with $\mathcal{O}(n^2)$ head movements. The same idea/algorithm shows that the problem is in $\mathcal{P}$.

For both problems, let $G = (V, E)$ and $|V| = n$. Then we know $|E| \in \mathcal{O}(n^2)$.

(b) We go through all possible 4-tuples (v_1, v_2, v_3, v_4) of V such that all the v_i's are different from each other. There exists $\binom{n}{4} \in \mathcal{O}(n^4)$ such 4-tuples. For each such 4-tuple (v_1, v_2, v_3, v_4), we examine whether each possible pair of that 4-tuple share an edge in E. Since $|E| \in O(n^2)$, this examination can be done in $O(n^2)$ time. If we found *one* 4-tuple that satisfies the requirement, we found a clique of size 4, the algorithm halts and accept. Otherwise, when we finish examining all possible 4-tuples, the algorithm halts and reject (since there is no clique of size at least 4 in G). The runtime of the above procedure is $\mathcal{O}(n^6)$, thus $4\text{-}\textsc{Clique} \in \mathcal{P}$.

(c) We test for all of the $\binom{|V|}{5} \in \mathcal{O}(n^5)$ possible subsets $C \subseteq V$ of size 5, whether they cover all edges.

To check if $C \subseteq V$ covers all edges we have to check for each edge $\{u, v\}$ whether $u \in C$ or $v \in C$, which takes at most $\in \mathcal{O}(|E||C|) \in \mathcal{O}(n^2)$ time. If we find a vertex cover (all edges covered for one subset C), then the algorithm halts and accepts. After all subsets were tested negatively, the algorithm halts and rejects (since there is no vertex cover of size at most 5 in G). The runtime of the above procedure is $\mathcal{O}(n^7)$, thus $5\text{-}\textsc{VertexCover} \in \mathcal{P}$.

Note: in case not stated that $n > 5$ in the problem definition, then one can add an extra check (of whether $n \leq 5$ or not) initially in the algorithm, which can still be done in polynomial time. Now, if $n \leq 5$, then the algorithm can directly halt and accept (since it is always true that all the nodes in any graph make up a vertex cover for that graph and hence in our case for $n \leq 5$, all the nodes of G is a vertex cover of size at most 5, so we know there exists one without the need of finding it), else it continues performing the algorithm above.
Similarly, for the 4-Clique problem, we can check intially if $n \leq 3$ or not, and if so, then the algorithm should directly halt and reject (since there will be no clique of size at least 4 in the graph anyway), else it continues performing the algorithm for the 4-Clique above.

Exercise 2: The Class $\mathcal{NP}$ *(Points)*

Show that the following problems (languages) are in class $\mathcal{NP}$.

(a) Given a graph $G = (V, E)$ and an integer k, it is required to determine whether G contains a clique of size at least k, hence consider the following problem:
Clique:= $\{\langle G, k\rangle \mid G$ has a clique of size at least k $\}$.

(b) A *hitting set* $H \subseteq \mathcal{U}$ for a given universe $\mathcal{U}$ and a set $S = \{S_1, S_2, \ldots, S_m\}$ of subsets $S_i \subseteq \mathcal{U}$, fulfills the property $H \cap S_i \neq \emptyset$ for $1 \leq i \leq m$ (H 'hits' at least one element of every S_i).

Given a universe set $\mathcal{U}$, a set S of subsets of $\mathcal{U}$, and a positive integer k, it is required to determine whether $\mathcal{U}$ contains a hitting set of size at most k, hence consider the following problem:
HittingSet:= $\{\langle \mathcal{U}, S, k\rangle \mid$ universe $\mathcal{U}$ has subset of size $\leq k$ that *hits* all sets in $S \subseteq 2^{\mathcal{U}}\}$.[1]

Sample Solution

We use the **Guess and Check** approach (and hence try building a polynomial time verifier) to show that the two problems are in $\mathcal{NP}$.

(a) Consider the following verifier for Clique on input $\langle\langle G, k\rangle, C\rangle$, that verifies in polynomial time that G has a clique of size at least k, where the idea of the certificate C is the clique. Let $|V| = n$. The verifier first tests if C has at least k different nodes from G with $O(|C| + |C| \cdot n + |C|^2)$ comparisons , then it tests whether $(u, v) \in E$ for every $u, v \in C$ in $O(|C|^2 \cdot n^2)$ comparisons. If both these tests pass, accept; else reject. Since the certificate has polynomial length in the input size, therefore the total running time is polynomial in the input size. So Clique has a polynomial time verifier. Therefore, Clique is in $\mathcal{NP}$.

(b) Consider the following verifier for HittingSet on input $\langle\langle \mathcal{U}, S, k\rangle, H\rangle$, that verifies in polynomial time that $\mathcal{U}$ has a hitting set of size at most k given S, where the idea of the certificate H is the hitting set. Let λ be the sum of the sizes of all the subsets S_i in S and δ the size of $\mathcal{U}$. Note that we can always check if A is a subset of B as we are doing before with the following brute-force algorithm: $\forall a \in A$ check if $\exists b \in B : a = b$ which needs $\mathcal{O}(|A| \cdot |B|)$ comparisons.
The verifier first tests whether H is a subset of $\mathcal{U}$ that has at most k elements with $\mathcal{O}(|H| + |H| \cdot \delta)$ comparisons. Then it tests whether it contains at least one element from each subset S_i in the collection S, with $\mathcal{O}(\lambda \cdot |H|)$ comparisons. If both tests pass, it accepts, and otherwise it rejects. Since the certificate has polynomial length in the input size, therefore the total running time is polynomial in the input size. So HittingSet has a polynomial time verifier. Therefore, HittingSet belongs in $\mathcal{NP}$.

[1] The power set $2^{\mathcal{U}}$ of some ground set $\mathcal{U}$ is the set of *all subsets* of $\mathcal{U}$. So $S \subseteq 2^{\mathcal{U}}$ is a collection of subsets of $\mathcal{U}$.