Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Philipp Bamberger, Yannic Maus

Theoretical Computer Science - Bridging Course
Summer Term 2018
Exercise Sheet 9

for getting feedback submit (electronically) before the start of the tutorial on 7th of January 2019.

Exercise 1: The class $\mathcal{NPC}$ *(8 Points)*

As this type of exercise is really important for the course we have another one.

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. Let

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

Show that $\textsc{DominatingSet}$ is in $\mathcal{NPC}$. Use that

$$\textsc{VertexCover} := \{\langle G, k\rangle \mid \text{ Graph } G \text{ has a } \textit{vertex cover} \text{ of size at most } k\} \in \mathcal{NPC} .$$

Remark: A $\textsc{VertexCover}$ *is a subset $V' \subseteq V$ of nodes of $G = (V, E)$ such that every edge of G is adjacent to a node in the subset.*

Exercise 2: Complexity Classes: Big Picture *(2+3+2 Points)*

(a) Why is $\mathcal{P} \subseteq \mathcal{NP}$?

(b) Show that $\mathcal{P} \cap \mathcal{NPC} = \emptyset$ if $\mathcal{P} \neq \mathcal{NP}$.
Hint: Assume that there exists a $L \in \mathcal{P} \cap \mathcal{NPC}$ and derive a contradiction to $\mathcal{P} \neq \mathcal{NP}$.

(c) Give a Venn Diagram showing the sets $\mathcal{P}, \mathcal{NP}, \mathcal{NPC}$ for both cases $\mathcal{P} \neq \mathcal{NP}$ and $\mathcal{P} = \mathcal{NP}$.
Remark: Use the results of (a) and (b) even if you did not succeed in proving those.

Exercise 3: The class $\mathcal{P}$ *(1+2+2+1 Points)*

CLIQUE:

- A *clique* of a graph $G = (V, E)$ is a subset $Q \subseteq V$ such that for all $u, v \in Q : \{u, v\} \in E$.
- **Input**: Encoding $\langle G, k \rangle$ of an undirected, unweighted, simple graph $G = (V, E)$ and $k \in \mathbb{N}$.
- **Question**: Is there a clique of size at least k?

$\mathcal{P}$ is the set of languages 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 ($\cong$ problems) 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) 1-DOMINATINGSET

(b) 2-VERTEXCOLORING

(c) 3-CLIQUE

(d) Any context-free language L. *Hint: You can use results from previous exercise sheets.*