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

Theoretical Computer Science - Bridging Course
Summer Term 2018
Exercise Sheet 8

for getting feedback submit (electronically) before the start of the tutorial on 17th of December 2018.

Exercise 1: The class P

Show that the following languages are in P.

(a) 5-CYCLE $= \{\langle G\rangle \mid G$ is a graph and contains a cycle of length 5$\}$.

Remark: A cycle of length 5 *in* G *are five distinct nodes* $v_0, \ldots, v_4$ *such that the edges* $\{v_i, v_{i+1 \mod 5}\}$, $i = 0, \ldots, 4$ *exist in* G.

(b) $L = \{a^n b^{3n} \mid n \geq 0\}$

(c) 17-INDEPENDENT SET $=\{\langle G\rangle \mid G$ is a graph and contains an independent set of size 17$\}$.

Remark: An independent set of a graph with size s *is a set* $S \subseteq V$, $|S| = s$ *such that* $\{v, w\} \notin E$ *for all* $u, w \in S$.

(d) Find a proper citation (e.g., via google) which states whether PRIMES$=\{\langle n\rangle \mid n \in \mathbb{N}$ is prime$\}$ is in P or not.

Repetition of Course Material *(0 Points)*

Let L_1, L_2 be languages (problems) over alphabets Σ_1, Σ_2. Then $L_1 \leq_p L_2$ (L_1 is polynomially reducible to L_2), iff a function $f : \Sigma_1^* \to \Sigma_2^*$ exists, that can be calculated in polynomial time and

$$\forall s \in \Sigma_1 : s \in L_1 \Longleftrightarrow f(s) \in L_2.$$

Language L is called $\mathcal{NP}$-hard, if *all* languages $L' \in \mathcal{NP}$ are polynomially reducible to L, i.e.

$$L\ \mathcal{NP}\text{-hard} \Longleftrightarrow \forall L' \in \mathcal{NP} : L' \leq_p L.$$

The reduction relation '$\leq_p$' is transitive ($L_1 \leq_p L_2$ and $L_2 \leq_p L_3 \Rightarrow L_1 \leq_p L_3$). Therefore, in order to show that L is $\mathcal{NP}$-hard, it suffices to reduce a known $\mathcal{NP}$-hard problem $\tilde{L}$ to L, i.e. $\tilde{L} \leq_p L$.
Finally a language is called $\mathcal{NP}$-complete ($\Leftrightarrow$: $L \in \mathcal{NPC}$), if

1. $L \in \mathcal{NP}$ and
2. L is $\mathcal{NP}$-hard.

Exercise 2: The Class $\mathcal{NPC}$ *(7 Points)*

This exercise (and similar ones) is really (!!) important for the course.

Show $\textsc{HittingSet} := \{\langle \mathcal{U}, S, k\rangle \mid \text{universe } \mathcal{U} \text{ has subset of size} \leq k \text{ that } \textit{hits} \text{ all sets in } S \subseteq 2^{\mathcal{U}}\} \in \mathcal{NPC}$.[1]

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 ***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).*
A ***vertex cover*** *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.*

Hint: For the poly. transformation ($\leq_p$) you have to describe an algorithm (with poly. run-time!) that transforms an instance $\langle G, k\rangle$ of VertexCover *into an instance $\langle \mathcal{U}, S, k\rangle$ of* HittingSet*, s.t. a vertex cover of size $\leq k$ in G becomes a hitting set of $\mathcal{U}$ of size $\leq k$ for S and vice versa(!).*

[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 $\mathcal{U}$.