Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
P. Schneider, C. Zheng
December 12, 2016

Theoretical Computer Science - Bridging Course
Winter Term 2016
Exercise Sheet 8

Hand in (electronically or hard copy) before your weekly meeting but not later than 23:59, Wednesday, December 21, 2016

Exercise 1: $\mathcal{O}$-Notation Formal Proofs (2+2+2 points)

The set $\mathcal{O}(f)$ contains all functions that are asymptotically not growing faster than the function f (when additive or multiplicative constants are neglected). That is:

$$g \in \mathcal{O}(f) \iff \exists c \geq 0, \exists M \in \mathbb{N}, \forall n \geq M : g(n) \leq c \cdot f(n)$$

For the following pairs of functions, check whether $f \in \mathcal{O}(g)$ or $g \in \mathcal{O}(f)$ or both. Proof your claims (you do not have to proof a negative result $\notin$, though).

(a) $f(n) = 100n,\ g(n) = 0.1 \cdot n^2$

(b) $f(n) = \log_2(n!),\ g(n) = n \log_2 n$ [*Hint:* $n! := \prod_{i=1}^{n} i \geq (n/2)^{n/2}$]

(c) $f(n) = 2^n,\ g(n) = 3^n$

Exercise 2: Sort Functions by Asymptotic Growth (4 points)

Sort the following functions by asymptotic growth. Write $g <_{\mathcal{O}} f$ if $g \in \mathcal{O}(f)$. Write $g =_{\mathcal{O}} f$ if $f \in \mathcal{O}(g)$ *and* $g \in \mathcal{O}(f)$.

n^2	$\sqrt{n}$	2^n	$\log(n^2)$
3^n	n^{100}	$\log(\sqrt{n})$	$(\log n)^2$
$\log n$	$10^{100}n$	$n!$	$n \log n$
$n \cdot 2^n$	n^n	$\sqrt{\log n}$	n

Exercise 3: Decision and Optimization Problems (2+2 points)

A *decision problem* is usually given as a formal question, whether a certain input (also called an *instance*) satisfies a condition or not. The language corresponding to a decision problem is the set of all instances encoded with symbols from an alphabet Σ, for which the answer to the question is **yes** and that are *well-formed* meaning that they represent a proper instance.
Quite often a problem can also be formulated as *optimization problem*, which asks for the maximum or minimum value that satisfies a certain condition. Algorithms solving decision problems can usually be used to solve the according optimization problem and vice versa. Consider the following problems:

DominatingSet:

- A *dominating set* of a graph $G = (V, E)$ is a subset $D \subseteq V$ such that for every vertex $v \in V$: $v \in D$ or v adjacent to a node $u \in D$.
- **Input**: Encoding $\langle G, k \rangle$ of an undirected, unweighted, simple graph $G = (V, E)$ and $k \in \mathbb{N}$.
- **Question**: Is there a dominating set with at most k nodes?

VertexColoring:

- A *vertex coloring* of a graph $G = (V, E)$ is a mapping $c : V \to \{1, \dots, k\}$ such that $c(u) = c(v) \Rightarrow \{u, v\} \notin E$.
- **Input**: Encoding $\langle G \rangle$ of an undirected, unweighted, simple graph $G = (V, E)$.
- **Question**: What is the smallest k for which a valid vertex coloring c of G exists?

(a) Are the above problems optimization or decision problems? Transform these problems into the respective other problem type.

(b) Give the languages k-DominatingSet and k-VertexColoring corresponding to decision problems of the above problems.

Exercise 4: 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.*