University of Freiburg
Institute for Computer Science
Prof. Dr. F. Kuhn

Exam Theoretical Computer Science - Bridging Course

Thursday, August 17, 2017, 10:00-12:00

Name: ....

Matriculation No.: ....

Signature: ....

Do not open or turn until told so by the supervisor!

- Write your **name** and **matriculation number** on this page and sign the document!
- Write your name on **all sheets**!
- Your **signature** confirms that you have answered all exam questions without any help, and that you have notified exam supervision of any interference.
- Write legibly and only use a pen (ink or ball point). Do **not use red**! Do **not use a pencil**!
- This is an **open book exam** therefore printed or hand-written material is allowed.
- However, **no electronic devices** are allowed.
- There are **eight tasks** (with several sub-tasks each) and there is a **total of 120 points**.
- **50 points are sufficient** in order to pass the exam.
- Only **one solution per task** is considered! Make sure to strike out alternative solutions, otherwise the one yielding the minimal number of points is considered.
- **Detailed steps** might help you to get more points in case your final result is incorrect.
- The keywords **Show...** or **Prove...** indicate that you need to prove or explain your answer carefully.
- The keywords **Give...** or **State...** indicate that you only need to provide a plain answer.
- You may use information given in a **Hint** without explaining them.
- **Read each task thoroughly** and make sure you understand what is expected from you.
- **Raise your hand** if you have a question regarding the formulation of a task.
- **Use the space below each task and the back of the sheet for your solution.** The last two sheets of this exam are blank and can be used for solutions. If you need additional sheets, raise your hand.

Question	1	2	3	4	5	6	7	8	**Total**
Points									
Maximum	14	17	15	14	15	13	16	16	120

Task 1: Basic Mathematical Skills (14 Points)

(a) Prove the equation $\sum_{i=0}^{n} 2^i = 2^{n+1} - 1$ for all $n \in \mathbb{N}_0$ **by induction** on n. *(5 Points)*

(b) Let A, B be sets. We define the symmetric difference $A \,\Delta\, B := (A \setminus B) \cup (B \setminus A)$.

Prove the following **implication**: *(4 Points)*

$$A \cap B \neq \emptyset \quad \Longrightarrow \quad A \,\Delta\, B \neq A \cup B.$$

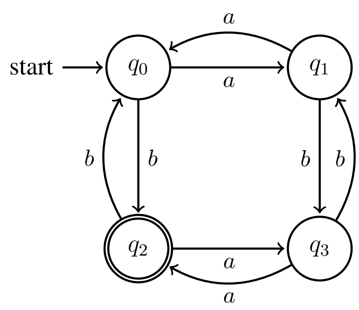
Remark: *$A \setminus B := A \cap \overline{B}$ is the 'set minus' operator, describing all elements of A that are not in B. Instead of a formal proof, you can show the implication with Venn diagrams.*

(c) Give the **minimum and the maximum number of edges** an undirected, **bipartite** graph $G = (V, E)$ with $n := |V|$ nodes can have. You may assume that n is even. *(1+4 Points)*

Remark: *A graph $G = (V, E)$ is **bipartite** if its nodes V can be partitioned into two disjoint sets $U, W \subseteq V$, such that there are **no** edges in E among any two nodes in U, and the same is true for W. That is, for all $\{v_1, v_2\} \in E$ it holds that v_1 and v_2 are not in the same part.*

Task 2: Regular Languages (17 Points)

Consider the following *Deterministic Finite Automaton (DFA)* A over the alphabet $\{a, b\}$.



(a) Give the **shortest string** accepted by A. *(1 Points)*

(b) Give an **infinite** set of strings that are accepted by A and **consist only of *b*'s**. *(3 Points)*

(c) **Describe the language** $L(A)$ recognized by A (as a set or verbally). *(4 Points)*

(d) Now consider the language $L(\gamma)$ given by the regular expression $\gamma := (ab)^*(ba)^*$.

Give a DFA that recognizes $L(\gamma)$ and has **at most four states**. *(6 Points)*

***Remark**: You can give a non-deterministic finite automaton (NFA) for a penalty of 2 points or an automaton with more than four states for a penalty of 1 point for each additional state.*

(e) Let L be the language consisting of words of the form $w_1w_2w_3$ with $w_1, w_2, w_3 \in \{a, b, c\}^*$ **and** w_1 contains **no** a's **and** w_2 contains **no** b's **and** w_3 contains **no** c's.

Give a regular expression that generates L. *(3 Points)*

Task 3: Context-Free Languages (15 Points)

Let $L = \{ss^{RC} \mid s \in \{0,1\}^*\}$ be a language over alphabet $\{0,1\}$, where s^{RC} describes the **reverse complement** of a string $s \in \{0,1\}^*$, obtained by reversing the order of symbols in s and then exchanging every 0 in s with 1 and every 1 in s with 0.

(a) State whether there is a string in L with an **unequal number of zeros and ones**. *(1 Points)*

(b) Give a **context-free grammar** that generates L. *(3 Points)*

(c) Give a **Pushdown Automaton** (PDA) that recognizes L. *(5 Points)*

(d) Prove that L is **not a regular language** by using the **Pumping Lemma**. *(6 Points)*

Task 4: Turing machines (14 Points)

(a) Give a **comparison** of the set of languages recognized by **deterministic** Turing machines with the set of languages recognized by **non-deterministic** Turing machines. *(2 Points)*

(b) State **two differences** between **deterministic** and **non-deterministic** Turing machines.

***Remark**: You obtain 1 point for the first difference and 2 points for the second. (1+2 Points)*

(c) One can define a variant of the Turing machine which allows **three** actions of the read/write-head: $\{\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{S}\}$, where $\boldsymbol{S}$ means that the head stands still during that step.

Let M_1 be a Turing machine **that uses head movements** $\{\boldsymbol{L}, \boldsymbol{R}, \boldsymbol{S}\}$. Give an **explicit** construction procedure that transfers M_1 into a Turing machine M_2 **that uses only head movements** $\{\boldsymbol{L}, \boldsymbol{R}\}$ and recognizes the same language, i.e. $L(M_1) = L(M_2)$. *(5 Points)*

(d) **Briefly** explain how to construct (or construct) a Turing machine for the language defined by the automaton depicted in Task 2 of this exam. *(4 Points)*

Task 5: $\mathcal{O}$ - Notation (15 Points)

State whether the following claims are true or false *(1 point each)*. Then **prove or disprove** the claim *(6 points for (a) and 7 points for (b))*. Use the definition of the $\mathcal{O}$-notation.

(a) $n^{\sqrt{2}} \in \mathcal{O}(\sqrt{2} \cdot n)$. ***Hint:*** $\sqrt{2} > 1$. *(1+6 Points)*

(b) $2^{\sqrt{n}} \in \mathcal{O}((\sqrt{2})^n)$. *(1+7 Points)*

Task 6: Decidability (13 Points)

(a) Consider the problem COLORING:

$$\text{COLORING} := \{\langle G, k\rangle \mid \text{undirected graph } G \text{ has a } \textbf{k-coloring}\}.$$

A **k-coloring** of $G = (V, E)$ is an assignment $c\colon V \to \{1, \dots, k\}$ of nodes to colors, such that no equally colored nodes are adjacent, i.e., for all edges $\{u, v\} \in E$ we have $c(u) \neq c(v)$.

(i) Show that COLORING is **decidable** by giving an algorithm (abstract description or pseudo-code) that decides whether a graph has a k-coloring. *(6 Points)*

(ii) Explain why your algorithm accepts **exactly** the instances $\langle G, k\rangle$ which have a k-coloring and why it always halts. *(2+1 Points)*

(b) Consider the problem MULTIPARTITION

$$\text{MULTIPARTITION} := \{\langle G, k\rangle \mid \text{undirected graph } G \text{ has a } \textbf{k-partition}\}.$$

A k-**partition** of $G = (V, E)$ is a partition of V into k disjoint subsets $V_1, \dots, V_k$ such that there are no edges among nodes from two different subsets. Formally: For all edges $\{u, v\} \in E$ it holds that u and v are in different subsets, i.e., $u \in V_i, v \in V_j$ with $i \neq j$.

A decider for COLORING can be used to show the decidability of MULTIPARTITION.

Explain how to use your algorithm for COLORING to decide MULTIPARTITION. *(4 Points)*

Remark*: If you did not succeed in giving an algorithm that decides* COLORING *in (a), you may assume that you have such an algorithm.*

Task 7: Complexity Theory (16 Points)

(a) Give a language which is **in $\mathcal{NP}$ but not in $\mathcal{P}$**. Assume that $\mathcal{P} \neq \mathcal{NP}$! *(2 Points)*

(b) Give a language which is **neither in $\mathcal{P}$ nor in $\mathcal{NP}$**. *(2 Points)*

(c) Given a set U of n elements ('universe') and a collection $S \subseteq 2^U$ of m subsets of U, a selection $C_1, \ldots, C_k \in S$ of k sets is called a **set cover** of size k if $C_1 \cup \ldots \cup C_k = U$. The SETCOVER-problem is defined as

$$\text{SETCOVER} := \{\langle U, S, k\rangle \mid U \text{ is a set, } S \subseteq 2^U \text{ and there is a \textbf{set cover} for } (U, S) \text{ of size } k\}.$$

Assume that we already know that the problem VERTEXCOVER is $\mathcal{NP}$-complete

$$\text{VERTEXCOVER} := \{\langle G, k\rangle \mid \text{undirected graph } G \text{ has a \textbf{vertex cover} of size at most } k\}.$$

Given a graph $G = (V, E)$, a **vertex cover** is a subset $V' \subseteq V$ of nodes of G such that every edge of G is adjacent to a node in the subset V'.

Show that SETCOVER is $\mathcal{NP}$-complete. *(12 Points)*

Hint: *For the polynomial reduction,* ***let the edges E*** *of a given instance of the* VERTEX-COVER *problem* ***be the universe U*** *for the associated instance of the* SETCOVER *problem.*

Task 8: Logic (16 Points)

(a) Consider the following propositional formula

$$\psi := (x \wedge y \to z \vee w) \wedge (y \to x) \wedge (z \wedge y \to 0) \wedge (w \wedge y \to 0) \wedge y.$$

(i) Transfer ψ into an equivalent formula in **conjunctive normal form (CNF)**. *(3 Points)*

(ii) Use the **resolution calculus** to show that ψ is unsatisfiable. *(5 Points)*

(b) Consider the following **first order logical** formulae

$$\begin{aligned} \varphi_1 &:= \forall x R(x,x) \\ \varphi_2 &:= \forall x \forall y\, R(x,y) \to (\exists z R(x,z) \wedge R(z,y)) \\ \varphi_3 &:= \exists x \exists y\, (\neg R(x,y) \wedge \neg R(y,x)) \end{aligned}$$

where x, y are variable symbols and R is a binary predicate. Give an interpretation

(i) I_1 which is a **model** of $\varphi_1 \wedge \varphi_2$. *(3 Points)*

(ii) I_2 which is **no model** of $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$. *(2 Points)*

(iii) I_3 which is a **model** of $\varphi_1 \wedge \varphi_2 \wedge \varphi_3$. *(3 Points)*

Remark*: No proof required.*