Theory of Computer Science (10948)
Model Exam

Spring semester 2019
University of Basel
Department of Mathematics and Computer Science

Name: ______________________________

Matriculation number: ______________________________

- The exam consists of a **multiple-choice part** and 6 **additional questions**.
- Please **write your name and your matriculation number on this title sheet**.
- You may prepare and use **one sheet of A4 paper with notes** (using both sides). Other aids such as lecture slides, notes, books, or calculators are not allowed. All electronic devices (such as mobile phones) must be switched off.
- You have **120 minutes** for working on the exam.
- For answering the questions, please use the space directly below each question. If you require more space, please use the reverse side of the same sheet.
- In case you develop several partial solutions for a question, please indicate clearly which one should be marked.

	Possible marks	**Marks achieved**
Multiple-Choice Questions	20	
Question 1	10	
Question 2	10	
Question 3	10	
Question 4	10	
Question 5	10	
Question 6	10	
Total	80	
Grade	(1.0–6.0)	

Multiple-Choice Questions (10× 2 marks)

(a) Which of the following general statements about propositional logic formulas φ and ψ are true?

- ☐ If φ is not valid, then φ is satisfiable.
- ☐ If $\varphi \models \psi$ and $\psi \models \varphi$ hold, then $\varphi \equiv \psi$.
- ☐ If $(\varphi \leftrightarrow \psi)$ is satisfiable then φ and ψ are logically equivalent.
- ☐ If $\varphi \equiv \psi$ and φ is valid then ψ is valid.

(b) Which of the following statements about propositional logic are true?

- ☐ If at least one of the sets of formulas Φ or Ψ is unsatisfiable, then $\Phi \cup \Psi$ is unsatisfiable.
- ☐ For every formula in DNF, there is a logically equivalent formula in CNF of the *same size*.
- ☐ If *KB* is an unsatisfiable knowledge base then one can derive the empty clause $\square$ with resolution from *KB*.
- ☐ If formula φ is unsatisfiable then it holds for every formula ψ that $\varphi \models \psi$.

(c) Which of the following statements about predicate logic are true?

- ☐ $(\forall x(P(x) \land \exists y(R(y, x) \lor \neg Q(y, x))) \lor P(y))$ is a sentence (= a closed formula).
- ☐ The variable assignment is irrelevant for determining the truth of a closed formula.
- ☐ $\forall x \forall y \mathrm{P}(x, y) \models \forall x \exists y \mathrm{P}(y, x)$
- ☐ $(\forall x \varphi \land \forall x \psi) \equiv \forall x(\varphi \land \psi)$

(d) Which of the following statements about regular languages are true?

- ☐ For every language that can be generated by a regular grammar, there exists a finite automaton (DFA, NFA) that accepts it.
- ☐ Every finite language is regular.
- ☐ The regular expression $a^*b^*|ab^*aa^*$ describes a language that contains the words ε and *abbbaa* but not word *abbaba*.
- ☐ The equivalence problem for regular languages is decidable.

(e) Which of the following statements about languages and automata are true?

- ☐ Regular languages cannot contain the empty word ε.
- ☐ There exist languages that cannot be accepted by any push-down automaton (PDA).
- ☐ For every context-free language, there is a deterministic Turing machine that accepts it.
- ☐ Every language that is accepted by some deterministic Turing machine is decidable.

(f) Which of the following statements are true? Please only consider *numerical* functions $f : \mathbb{N}_0^k \to_p \mathbb{N}_0$, not functions with *words* as input.

- ☐ WHILE programs are more powerful than GOTO programs.
- ☐ For every WHILE program, it is possible to construct a LOOP program that computes the same function.
- ☐ Every total function is LOOP-computable.
- ☐ Every primitive recursive function can be computed by a LOOP program.

(g) Let X be an undecidable problem. Which of the following statements follow from this?

- ☐ X is semi-decidable.
- ☐ X is not semi-decidable.
- ☐ All problems Y with $X \leq Y$ are undecidable.
- ☐ All problems Y with $Y \leq X$ are undecidable.

(h) Which of the following problems/languages are decidable?

- ☐ Does a given GOTO program terminate if all input parameters are 0?
- ☐ The language $\bar{L}$, where L is decidable.
- ☐ Does a given graph have a vertex cover of size at most K?
- ☐ Does a given WHILE program compute a given μ-recursive function?

(i) Let A, B, C be problems in NP with $A \leq_p B$ and $B \leq_p C$ and B being NP-hard. Which of the following statements can you derive from this?

- ☐ A is NP-complete.
- ☐ A is not NP-complete.
- ☐ B is NP-complete.
- ☐ C is NP-complete.

(j) Let X be a problem in P and let Y be an NP-complete problem. Which statements follow?

- ☐ There exists a deterministic polynomial algorithm for X.
- ☐ If $X \leq_p Y$ holds then P = NP.
- ☐ If there exists a deterministic polynomial algorithm for Y, then there exists a deterministic polynomial algorithm for SAT.
- ☐ Y is decidable.

Question 1 (4 + 4 + 2 marks)

(a) Transform the following formula into DNF by applying equivalence rules. For each step, only apply one equivalence rule.

$$\psi = (\neg(A \to \neg(B \lor (D \land E))) \to C)$$

(b) Prove without a truth table or applications of equivalence rules that

$$\varphi = (((A \lor B) \to (B \land C)) \to (\neg A \lor B))$$

is a tautology. If you want, you can replace the implications with the formulas they abbreviate. Alternatively you can directly argue with the corresponding semantics of the implications.

(c) Specify a model with universe $U = \{a, b, c, d\}$ for the following predicate logic formula for signature $\langle\{x, y, z\}, \{\mathrm{k}\}, \{\}, \{P\}\rangle$, where P has arity 3.

$$\chi = \forall x \exists y P(x, y, \mathrm{k})$$

Additional space for question 1:

Question 2 (7 + 3 marks)

(a) Use the pumping lemma to prove that the language

$$L = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0, i = j + k\}$$

is not regular.

(b) Specify a DFA that accepts the language that is described by the regular expression $\gamma = \mathtt{0}^*\mathtt{1}(\mathtt{01})^*$.

Additional space for question 2:

Question 3 (5 + 5 marks)

Consider language $L = \{\mathtt{a}^i\mathtt{b}^j\mathtt{c}^k \mid i, j, k \geq 0, j = i + k\}$.

(a) Specify a context-free grammar that generates L.

(b) Specify a push-down automaton (PDA) that accepts L.

Specify for both parts a complete description with all components.

Additional space for question 3:

Question 4 (4 + 1 + 5 marks)

(a) Which function f is produced by applying the primitive recursion scheme to the following two functions? Describe f as simply as possible.

$$g(a) = 0$$
$$h(a, b, c) = 2a + c$$

(b) Is f WHILE-computable? Is f μ-recursive? Is f LOOP-computable? Is f Turing-computable?

(c) Simulate the syntactical construct

$$\textbf{IF } x_i < c \textbf{ THEN } P \textbf{ END}$$

(with $i, c \in \mathbb{N}_0$, P a LOOP program) with a LOOP program that only uses the constructs from the basic definition.

Reminder: LOOP programs are inductively defined as follows:

- $x_i := x_j + c$ is a LOOP program for every $i, j, c \in \mathbb{N}_0$
- $x_i := x_j - c$ is a LOOP program for every $i, j, c \in \mathbb{N}_0$
- If P_1 and P_2 are LOOP programs, then so is $P_1; P_2$
- If P is a LOOP program, then so is LOOP x_i DO P END for every $i \in \mathbb{N}_0$

Note that there is *no* statement $x_i := c - x_j$ with $i, j, c \in \mathbb{N}_0$.

Additional space for question 4:

Question 5 (4 + 6 marks)

(a) In each part, give an example of a language L_i with the given properties (without justification), or explain why no such language exists (with a short explanation).

1. L_1 is undecidable and L_1 and $\overline{L_1}$ are semi-decidable.
2. L_2 is a type-0 language and decidable.
3. L_3 is a type-0 language and undecidable.
4. L_4 is in NP and undecidable.

(b) Which of the following informally described algorithmic problems are computable? Give brief justifications (1 sentence each).

1. Given two NFAs M and M', is there an input w that is accepted by both M and M'?
2. Given a deterministic Turing machine M, compute a WHILE program that computes the same function as M.
3. Given a WHILE program P and a GOTO program P', do P and P' compute the same function?

Additional space for question 5:

Question 6 (4 + 6 marks)

Consider the following decision problems:

HittingSet:

- *Given:* finite set T, set of sets $S = \{S_1, \dots, S_n\}$ with $S_i \subseteq T$ for all $i \in \{1, \dots, n\}$, natural number $K \in \mathbb{N}_0$
- *Question:* Is there a set H with at most K elements, which contains at least one element from each set in S.

  Formally: Is there a set H with $|H| \leq K$ and $H \cap S_i \neq \emptyset$ for all $i \in \{1, \dots, n\}$?

VertexCover:

- *Given:* undirected graph $G = \langle V, E \rangle$, natural number $K \in \mathbb{N}_0$
- *Question:* Does G have a vertex cover of size at most K, i.e.,
  a set of vertices $C \subseteq V$ with $|C| \leq K$ and $\{u, v\} \cap C \neq \emptyset$ for all $\{u, v\} \in E$?

(a) Show HittingSet $\in$ NP by specifying a non-deterministic polynomial algorithm.

(b) Prove that HittingSet is NP-hard. You may use that VertexCover is NP-complete.

Additional space for question 6: