Theory of Computer Science (10948-01)
Practice Exam Spring Term 2015

Prof. Dr. Malte Helmert
Gabriele Röger
University of Basel
Department of Mathematics and Computer Science

Name: ______________________________

Immatriculation number: ______________________________

- The exam consists of a **multiple-choice part** and 6 **additional questions**.
- Please **write your name and your immatriculation 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 **105 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 Punkte)

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

- ☐ If φ is satisfiable then $\neg\varphi$ is unsatisfiable.
- ☐ If φ is unsatisfiable then it holds for every ψ that $\varphi \models \psi$.
- ☐ For every formula there is a logically equivalent CNF formula of the same size.
- ☐ With resolution, we can derive the empty clause $\square$ from any unsatisfiable knowledge base that is given as a set of clauses.

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

- ☐ $(\forall x(P(x) \land \exists y(Q(y,x) \lor (x = y))) \lor P(y))$ is a sentence.
- ☐ $(\forall x\varphi \land \forall x\psi) \equiv \forall x(\varphi \land \psi)$
- ☐ $(\forall x\varphi \lor \forall x\psi) \equiv \forall x(\varphi \lor \psi)$
- ☐ If the sets Φ and Ψ of formulas are both satisfiable then also the set $\Phi \cup \Psi$ is satisfiable.

(c) Which of the following statements about languages and grammars are true?

- ☐ For every language there is a grammar that generates it.
- ☐ Every context-free language can be generated by a context-sensitive grammar.
- ☐ For $\Sigma = \{\mathtt{a}, \mathtt{b}\}$ it holds that $\varepsilon \in \Sigma^*$ and $\mathtt{aba} \in \Sigma^*$.
- ☐ Every infinite Type-0 language contains the empty word ε.

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

- ☐ Every finite language is regular.
- ☐ For every NFA with n states there is an NFA with $n + 1$ states but only one accepting state that recognizes the same language.
- ☐ If an NFA with n states recognizes a language then the minimal DFA for this language has at most n states.
- ☐ With the pumping lemma one can prove that a language is regular.

(e) Which of the following statements about context-free languages and PDAs are true?

- ☐ Every language that can be generated by a context-free grammar can be recognized by a PDA.
- ☐ The language $L = \{\mathtt{a}^n\mathtt{b}^n \mid n \in \mathbb{N}_0\}$ is context-free.
- ☐ Every context-free language can be recognized by a PDA with only one state.
- ☐ If L_1 is a context-free language and L_2 is a regular language then $L_1 \cup L_2$ is context-free.

(f) Which of the following statements are true? For this question, only consider *numerical* functions $f : \mathbb{N}_0^k \to \mathbb{N}_0$ and no functions with *words* as inputs.

- ☐ Turing machines are less powerful than WHILE programs.
- ☐ For every deterministic Turing machine, it is possible to construct a GOTO program computing the same function.
- ☐ For every GOTO program, it is possible to construct a deterministic Turing machine computing the same function.
- ☐ The Ackermann function is WHILE-computable.

(g) Which of the following problems are decidable?

- ☐ "Does a given GOTO program halt if all input variables contain the value 0?"
- ☐ the language $L \cup \bar{L}$, where L is semi-decidable
- ☐ the Travelling Salesperson Problem (TSP)
- ☐ the language $\{\varepsilon\}$

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

- ☐ All problems Y with $X \leq Y$ are undecidable.
- ☐ All problems Y with $Y \leq X$ are undecidable.
- ☐ X is the Halting Problem.
- ☐ X and $\bar{X}$ are semi-decidable.

(i) Which of the following statements describe the proof idea of the Cook-Levin theorem?

- ☐ Translate the workings of a nondeterministic Turing machine with polynomial runtime into a logical formula.
- ☐ Show that the satisfiability problem of propositional logic is undecidable.
- ☐ Reduce every problem in NP to SAT.
- ☐ For every polynomial-size logic formula, construct a Turing machine that satisfies it.

(j) Let X and Y be NP-complete problems. Which of these statements follow?

- ☐ If there exists an efficient algorithm for X, then there also exists an efficient algorithm for Y.
- ☐ If there exists an efficient algorithm for X, then there also exist efficient algorithms for BinPacking and TSP.
- ☐ $X \leq_{\mathrm{p}}$ SAT
- ☐ SAT $\leq_{\mathrm{p}} X$

Question 1 (4+2+4 marks)

(a) Specify a model for the following propositional logic formula and use the semantics of propositional logic to prove that it indeed satisfies the formula.

$$((A \land (B \lor C)) \land \neg C)$$

(b) For each of the following properties, specify a propositional formula over atoms $\{A, B, C\}$ that has this property.

satisfiable:	
valid:	
unsatisfiable:	
has exactly 3 models:	

(c) Show that the following propositional formulas are equivalent with the equivalences from the lecture.

$$((B \lor \neg C) \rightarrow (A \lor B)) \equiv ((A \lor B) \lor C)$$

Additional space for question 1:

Question 2 (4+6 marks)

Let $\Sigma = \{\mathtt{a}, \mathtt{b}\}$.

(a) Specify a DFA that accepts the language described by the regular expression $\mathtt{a}^*\mathtt{b}(\mathtt{ab})^*$. It is sufficient to specify the DFA graphically with a diagram.

(b) Use the Pumping Lemma to show that $L = \{\mathtt{a}^n\mathtt{b}^{2n} \mid n \geq 0\}$ is not regular.

Additional space for question 2:

Question 3 (4+6 marks)

Consider the following language

$$L = \{\mathtt{a}^n \mathtt{b}^m \mathtt{c}^{2n} \mid n, m \geq 0\}$$

over the alphabet $\Sigma = \{\mathtt{a}, \mathtt{b}, \mathtt{c}\}$.

(a) Specify a context-free grammar G that generates L, i.e., $\mathcal{L}(G) = L$. Specify all components of G.

(b) Construct a push-down automaton (PDA) M which accepts exactly L. It is sufficient to specify the PDA graphically with a diagram.

Reminder: a PDA is initialized with the stack symbol `#` on the stack. It accepts an input if and only if the input word has been processed completely and the stack is empty (i.e., there is no accepting state).

Additional space for question 3:

Question 4 (4+4+2 marks)

(a) Write a LOOP program that calculates the sum of the first n natural numbers, i.e., it calculates the following function:

$$f(n) = \sum_{i=1}^{n} i$$

You may use all syntax constructs from the lecture and the exercises.

(b) Simulate the following syntax construct with known constructs for LOOP programs. Its semantics are as follows: x_i is incremented from 1 to x_j and P is executed once for each of these values. You may use all syntax constructs from the lecture and the exercises. Also, you may assume that x_i and x_j are not changed in P.

$$
\begin{array}{l}
\textbf{FOR } x_i = 1 \textbf{ TO } x_j \textbf{ DO} \\
\quad P \\
\textbf{END}
\end{array}
$$

(c) Which unary function does the following WHILE program calculate? Is it possible to calculate the same function with a LOOP program? Please justify your answer to the second question.

$$
\begin{array}{l}
x_2 := 1; \\
x_3 := 0; \\
\textbf{WHILE } x_2 \neq 0 \textbf{ DO} \\
\quad \textbf{IF } x_1 = x_3 \textbf{ THEN} \\
\quad\quad x_2 := 0 \\
\quad \textbf{END}; \\
\quad x_3 := x_3 + 2 \\
\textbf{END}; \\
x_0 := 1
\end{array}
$$

Additional space for question 4:

Question 5 (3+1+6 marks)

(a) Describe informally (1 sentence per property) what it means that a given language $L \subseteq \Sigma^*$ is

- *decidable*,
- *semi-decidable*,
- *undecidable*.

(b) Describe (without proof) the relationships between the properties *decidable*, *semi-decidable*, *undecidable*: which properties imply which other properties? Which properties are mutually exclusive?

(c) Which of the following languages are decidable? Give brief justifications (1 sentence per language).

- $L_1 = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid M_w \text{ computes a function with a finite domain}\}$
- $L_2 = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid M_w \text{ computes a LOOP-computable function }\}$
- $L_3 = \{w \in \{\mathtt{0}, \mathtt{1}\}^* \mid M_w \text{ computes a Turing-computable function }\}$

Hint: Use Rice's theorem (where possible) to show undecidability.

Additional space for question 5:

Question 6 (3+7 marks)

Consider the following decision problems:

DirHamiltonPath:

- *Given:* Directed graph $G = \langle V, E \rangle$
- *Question:* Does G contain a Hamiltonian path?

DirHamiltonPathWithEndPoints:

- *Given:* Directed graph $G = \langle V, E \rangle$, start node $v_\mathrm{s} \in V$, end node $v_\mathrm{e} \in V$
- *Question:* Does G contain a Hamiltonian path from v_s to v_e,
  i.e., a Hamiltonian path $\pi = \langle v_1, \dots, v_n \rangle$ with $v_1 = v_\mathrm{s}$ and $v_n = v_\mathrm{e}$?

(a) Show that DirHamiltonianPathWithEndPoints is in NP
by specifying a non-deterministic, polynomial algorithm.

(b) Prove that DirHamiltonianPathWithEndPoints is NP-hard.
You may use that the problem DirHamiltonianPath is NP-complete.

Reminder: A *Hamiltonian path* in a directed graph $\langle V, E \rangle$ is a sequence of nodes $\pi = \langle v_1, \dots, v_n \rangle$ that defines a path ($\langle v_i, v_{i+1} \rangle \in E$ for all $1 \leq i < n$) and contains each node of V exactly once.

Additional space for question 6: