Computability and Complexity
Exam January 31, 2020

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

Family Name:

Given Name:

Matriculation Number:

Study Code:

Total: 100 Points.

$\geq$ 50 Points: GEN4
$\geq$ 63 Points: BEF3
$\geq$ 75 Points: GUT2
$\geq$ 88 Points: SGT1

Please write on the empty sheets/back pages; you may also add additional pages.

1. (20P) Let L be the language over the alphabet $\{0, 1\}$ whose words contain 00 and 11 (e.g., 0010111 and 01101001 are in L, but not 001 and not 0101).
    a) (5P) Give a regular expression for L.
    b) (6P) Define a non-deterministic (not a deterministic) finite state machine $(Q, \Sigma, \delta, S, F)$ whose language is L (the transition function by both a table and a graph).
    c) (9P) Define a deterministic finite state machine whose language is L (the transition function by both a table and a graph).

2. (15P) Give a finite state machine over the alphabet $\{a, b, c, d, e\}$ whose language is the language of the following regular expression:

$$(a + (b \cdot c)^* \cdot d)^* + e$$

   It suffices to depict the machine by a graph; show also the crucial intermediate steps of the construction, not only the final result.

3. (10P) Let L be the language of all bit strings that have at most 5 occurrences of bit 1 directly after each other and at most 5 occurrences of bit 0 directly after each other (i.e., 000 and 01111101 are in L but 1000000 is not). Is L regular or not? Justify your answer in detail (hint: the justification does *not* demand the construction of a concrete state machine for the language).

4. (10P) Is the problem "Does Turing-machine M accept any input?" (i.e., "Is there some word w such that M accepts w?") semi-decidable? Is it decidable? Justify both of your answers in detail, e.g., by a construction of a Turing machine that takes input $\langle M \rangle$ and (semi-)decides the problem or by arguing why such a machine does not exist.

5. (10P) Take the function $f : \mathbb{N} \times \mathbb{N} \to \mathbb{N}$ defined as

$$f(y, x) := \begin{cases} 0 & \text{if } y + y \geq x \\ 1 & \text{otherwise} \end{cases}$$

   What does the function $(\mu f) : \mathbb{N} \to \mathbb{N}$ compute? Is (μf) LOOP-computable? If not, argue why not. If yes, give a LOOP program that computes (μf) (in such a program you may freely use the addition function, the greater-equal predicate, and the conditional statement, all of which are clearly LOOP-computable).

6. (10P) Is the following statement true or not?

$$\sqrt{3n^2 + 5} = O(n)$$

   Justify your answer by a proof (possibly by contradiction) using the definition of O.

7. (15P) Take the Java function

```
static int f(int a, int b) {
  if (a+1 >= b) return 1;
  int s = 1;
  s = s+f(a,(a+b)/2);
  s = s+f((a+b)/2,b);
  return s;
}
```

Let $T(n)$ denote the total number of calls of f in the execution of $f(a, b)$ where $n = b - a$; you may assume $n = 2^m$ for some $m \in \mathbb{N}$.

a) (3P) Sketch the recursion tree for the execution of $f(0, 16)$ and compute $T(16)$.

b) (3P) Give a definition of $T(n)$ as a recurrence (do not forget the base case).

c) (3P) Give an explicit solution of $T(n)$ by a sum of the nodes in each tree level.

d) (3P) Give an explicit solution of $T(n)$ by a closed formula.

e) (3P) Give an asymptotic estimation $T(n) = \Theta(\ldots)$.

8. (10P) Are the following statements true or not? Justify your answers.

a) (5P) If a nondeterministic Turing-machine can decide in polynomial time whether an arbitrary propositional formula is satisfiable, then $\mathcal{P} = \mathcal{NP}$.

b) (5P) If $\mathcal{P} = \mathcal{NP}$, then every problem that can be decided by a deterministic Turing machine in exponential time can also be decided by a deterministic Turing machine in polynomial time.