Computability and Complexity
Sample Exam Questions

Wolfgang Schreiner
Wolfgang.Schreiner@risc.jku.at

Family Name:

Given Name:

Matriculation Number:

Study Code:

Total: 100 Points.

≥ 51 Points: GEN4
≥ 64 Points: BEF3
≥ 77 Points: GUT2
≥ 89 Points: SGT1

1. (20P) Let L be the language over the alphabet $\{0, 1\}$ whose words contain the string 110 at least twice, namely at the beginning and at the end of the word.

   a) (4P) Give a regular expression that denotes L.

   b) (6P) Define a non-deterministic finite state machine $M = (Q, \Sigma, \delta, S, F)$ whose language is L (the transition function by both a table and a graph).

   c) (6P) Define a deterministic finite state machine whose language is L.

   d) (4P) Define a finite state machine whose language is the complement of L.

2. (10P) Construct a non-deterministic finite state machine whose language is denoted by the regular expression $1 + (2^* + 1 \cdot 2^* \cdot 3 \cdot (2 + 3)^*)^*$.

3. (16P) Which of the following languages over the alphabet $\{0, 1\}$ are regular? Justify your answers in detail.

   a) (4P) $L_a := \{0^m : \ m \in \mathbb{N} \wedge 2|m \wedge 3|m\}$

   b) (4P) $L_b := \{0^m 1^n : \ m, n \in \mathbb{N} \wedge 2|m \wedge 3|n\}$

   c) (4P) $L_c := \{0^m 1^n : \ m, n \in \mathbb{N} \wedge m|n\}$

   d) (4P) $L_d := \{0^m 1^n : \ m, n \in \mathbb{N} \wedge m|n \wedge n < 1000\}$

4. (16P) Let $\langle M \rangle$ denote the code of a Turing machine M with alphabet $\{0, 1\}$. Which of the following languages are recursively enumerable and/or recursive? Justify your answers in detail.

   a) (4P) $L_1 := \{\langle M \rangle : 10101 \in L(M)\}$

   b) (4P) $L_2 := \{\langle M \rangle : 10101 \notin L(M)\}$

   c) (4P) $L_3 := \{\langle M \rangle : L(M)$ ist recursively enumerable$\}$

   d) (4P) $L_4 := \{\langle M \rangle : L(M)$ ist recursive$\}$

5. (10P) Are the following statements true or not? Justify your answers in detail.

   a) (5P) If L_1 and L_2 are recursive languages, then also their difference $L_1 \setminus L_2$ is recursive.

   b) (5P) If L_1 and L_2 are recursively enumerable, then also their difference is recursively enumerable.

6. (15P) Take the LOOP program with inputs x_1, x_2 and output x_0

```
x_0 := 0
x_3 := x1
loop x_2-x_1+1 do
  if x_3 is prime and x_0 = 0 then x_0:= x_3 end
  x_3 := x_3+1
end
```

which computes the smallest prime x_0 with $x_1 \leq x_0 \leq x_2$ (respectively $x_0 = 0$, if no such prime exists) using an auxiliary loop program "x_3 *is prime and* $x_0 = 0$". Give a primitive recursive definition of the same function (or argue why this is not possible).

Hint: we know that for every primitive recursive function p, the function f defined as

$$f(\ldots) := \begin{cases} \ldots, & \text{if } p(\ldots) = \ldots \\ \ldots, & \text{else} \end{cases}$$

is also primitive recursive.

7. (10P) Formally prove or disprove $5n^2 + 7 = O(2^n)$.

8. (10P) Formally prove $\sum_{k=1}^{n} k^2 = n \cdot (n+1) \cdot (2n+1)/6$.

9. (10P) Formally prove that for all $n = 2^m$, the recurrence $T(1) = 1, T(n) = 4 \cdot T(n/2)$ is solved by $T(n) = n^2$.

10. (15P) Let $T(n)$ be the number of times that the command C is executed in the following program.

```
for (i=0; i<n; ++)
  for (j=i+1; j<n; j++)
    for (k=i; k<j; k++)
      C;
```

    a) Compute $T(4)$.

    b) Give an explicit definition of $T(n)$ by a nested sum and derive from this an asymptotic estimation $T(n) = \Theta(\ldots)$.

    c) Give an explicit definition of $T(n)$ with the help of a single summation symbol (hint: first consider which pairs of i, j may arise in the program, then consider how often the innermost loop is executed for each pair, then consider how often the same iteration number for the innermost loop occurs for all pairs).

    d) Give an explicit definition of $T(n)$ without using a summation symbol.

11. (10P) Consider two programs with the following shape

```
P1(a, b, ...):
  n = b-a;
  for (i=0; i<n; i++)
    for (j=i; j<n; j++)
      ...
  return ...
```

```
P2(a, b, ...):
n = b-a;
if (n <= 0) return ...;
for (i=0; i<7; i++) {
  c = ...i...;
  ... P(c, c+n/3) ...;
}
return ...
```

where the parts marked as "..." are executed in time $O(n)$.

Which program runs faster for large input measures? Justify your answer in detail.

12. (10P) Is the following problem semi-decidable by a Turing machine? If yes, give an informal construction of this machine (pseudo-code and/or diagram plus explanation). If not, then justify your answer in detail.

    Decide for given Turing machine codes $\langle M_1 \rangle$ and $\langle M_2 \rangle$, whether $L(M_1) \cap L(M_2) \neq \emptyset$.

    Answer the question also for the problem $L(M_1) \cap L(M_2) = \emptyset$.

13. (12P) Are these statements true or not? Justify your answers in detail.
    a) (4P) The function $s(n) := \sum_{i=1}^{n} i$ is μ-recursive..
    b) (4P) For every primitive-recursive function f, the function $t(n, m) := \min i : n \leq i \leq m \wedge f(i) = 1$ is primitive recursive.
    c) (4P) For every primitive-recursive function f, the function

    $$t(n,m) := \begin{cases} 0 & \text{if } \forall i : n \leq i \leq m \Rightarrow f(i) = 0 \\ \min i : n \leq i \leq m \wedge f(i) = 1 & \text{else} \end{cases}$$

    is primitive recursive.

14. (10P) Are these statements true or not? Justify your answers in detail.

    Let L be the set of strings of form '$1^n + 1^m = 1^{n+m}$'' (e.g. "111+11=11111" is in L).
    a) (5P) There exists a regular expression R with $L(R) = L$.
    b) (5P) There exists a grammar G with $L(G) = L$.

15. (28P) Are these statements true or not? Justify your answers in detail.
    a) (4P) If both $f : \{0\}^*\{0\}^* \rightarrow$ and $g : \{0\}^* \rightarrow \{0\}^*$ are computable in polynomial time, then also $f \circ g$ is.
    b) (4P) If $\mathcal{P} = \mathcal{NP}$ holds, then every problem that can be deterministically solved in exponential time can be also solved in polynomial time.
    c) (4P) If there is a problem in $\mathcal{NP}$ that can be also solved deterministically in polynomial time, then $\mathcal{P} = \mathcal{NP}$.
    d) (4P) If a problem P is decidable by a deterministic Turing machine, then also its complement is.
    e) (4P) If a problem P is decidable by a deterministic Turing machine in polynomial time, then also its complement is.
    f) (4P) If a problem P is decidable by a nondeterministic Turing machine, then also its complement is.
    g) (4P) If a problem P is decidable by a nondeterministic Turing machine in polynomial time, then also its complement is.