Albert-Ludwigs-Universität, Inst. für Informatik
Prof. Dr. Fabian Kuhn
Philipp Bamberger, Yannic Maus

Theoretical Computer Science - Bridging Course
Summer Term 2018
Exercise Sheet 5

for getting feedback submit (electronically) before the start of the tutorial on 26th of November 2018.

Exercise 1: Constructing a Turing Machine *(3 Points)*

Consider alphabet $A = \{1, 2, \ldots, 9\}$. We call a string S over A a *blue* string, if and only if the string consisting of the odd-positioned symbols in S is the reverse of the string consisting of the even-positioned symbols in S. For example $S = 14233241$ is a blue string since the substring of the odd-positioned symbols is 1234 which is the reverse of the substring of the even-positioned symbols, i.e., 4321.
Design a Turing machine which accepts all blue strings over A. You do not need to provide a formal description of the Turing machine but your description has to be detailed enough to explain every possible step of a computation.

Exercise 2: *(4+2+2 Points)*

(a) Design a Turing Machine that decides the language $L := \{0^n 1^n \mid n \geq 1\}$. Explain your choice (you are supposed to explicitly construct the Turing machine).

(b) Give the sequence of configurations of your Turing machine run on the string 0011.

(c) Give the sequence of configurations of your Turing machine run on the string 0010.

Remark: Here, you need to solve part a) to solve part b) and c). We would try to avoid such exercises in the exam.

Exercise 3: Random Questions *(2+2 Points)*

(a) Does the fact that the Halting Problem is not decidable mean that we can never tell if a program we have written is going to halt? Explain.

(b) Describe how a Turing machine with arbitrary tape alphabet Γ_0 can be simulated by a Turing machine with tape alphabet $\Gamma_1 = \{0, 1, \square\}$ that never writes the symbol $\square$ on the tape.

Exercise 4: PDA to Turing Machine *(10 Points)*

Let a k-PDA be a pushdown automaton that has k stacks. Thus a 0-PDA is an NFA and a 1-PDA is a conventional PDA. We already know that 1-PDAs are more powerful (recognize a larger class of languages) than 0-PDAs.

(a) (5 points) Show that 2-PDAs are more powerful than 1-PDAs. *Hint: Find a suitable language that cannot be recognized by a* 1*-PDA but can be recognized by a* 2*-PDA*

(b) (5 points) Show that 3-PDAs are not more powerful than 2-PDAs. *Hint: Simulate a Turing machine tape with two stacks.*