University of Freiburg
Dept. of Computer Science
Prof. Dr. F. Kuhn
S. Faour, P. Schneider

Theoretical Computer Science - Bridging Course
Exercise Sheet 6

Due: Wednesday, 9th of June 2021, 12:00 pm

Exercise 1: Proving Decidability *(6 Points)*

Show that the following languages are decidable.

- Let $A = \{\langle M \rangle \mid M$ is a DFA which doesn't accept any string containing an odd number of 1s$\}$.
- Let $B = \{\langle G \rangle \mid G$ is a CFG over $\{0,1\}$ and $1^* \cap L(G) \neq \phi\}$. Use the fact that a language $C \cap R$ is context free for some context free language C and regular language R (bonus try to show it).

Remark: You can use that it is not difficult to construct a TM, which tests whether an input is the well formed encoding of a DFA.

Exercise 2: Proving Undecidability *(8 Points)*

Show that both the halting problem and its special version are both undecidable.

- The *halting problem* is defined as
$$H = \{\langle M, w \rangle \mid \langle M \rangle \text{ encodes a TM and } M \text{ halts on string } w\}.$$
Hint: Assume H is decidable and try to reach a contradiction by showing some known undecidable problem (cf. lecture) decidable .
- The *special halting problem* is defined as
$$H_s = \{\langle M \rangle \mid \langle M \rangle \text{ encodes a TM and } M \text{ halts on } \langle M \rangle\}.$$
Hint: Assume that M is a TM which decides H_s and then construct a TM which halts iff M does not halt. Use this construction to find a contradiction.

Exercise 3: Semi-Decidable vs. Recursively Enumerable *(6 Points)*

Very often people in computer science use the terms *semi-decidable* and *recursively enumerable* equivalently. The following exercise shows in which way they actually are equivalent. We first recall the definition of both terms.

A language L is *semi-decidable* if there is a Turing machine which accepts every $w \in L$ and does not accept any $w \notin L$ (this means the TM can either reject $w \notin L$ or simply not stop for $w \notin L$).

A language is *recursively enumerable* if there is a Turing machine which eventually outputs every word $w \in L$ and never outputs a word $w \notin L$.

(a) Show that any recursively enumerable language is semi-decidable.

(b) Show that any semi-decidable language is recursively enumerable.