Tutorial 4

Exercise 1 (compulsory)

Consider the following two claims:

Claim 1: Every language L which is a subset of A_{TM} ($L \subseteq A_{TM}$) is undecidable.

Claim 2: Every language L which is a superset of A_{TM} ($A_{TM} \subseteq L$) is undecidable.

Which of these claims are true? Provide the right arguments or give counter-examples.

Exercise 2 (compulsory)

Which of the following languages are decidable? If you claim that a particular language is decidable, provide a decider for the language.

- $L_1 \stackrel{\text{def}}{=} \{\langle A, B\rangle \mid A$ and B are DFA and $L(A) \cap L(B) = \emptyset \}$
- $L_2 \stackrel{\text{def}}{=} \{\langle M\rangle \mid M$ is a TM and M contains more than 5 states $\}$
- $L_3 \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid M$ is a TM and M accepts w in less than 1000 computational steps $\}$
- $L_4 \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid M$ is a TM and M accepts w in a finite number of computational steps $\}$
- $L_5 \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid M$ is a TM and M on w halts either in accept or reject state$\}$

Exercise 3 (compulsory)

In the proof of Theorem 4.11 we construct a machine D and reach a contradiction by running D on the input $\langle D\rangle$. Why could we not instead of constructing D continue the proof as follows?

Construct the machine D_1 :

D_1 = "On input $\langle M\rangle$, where M is a TM:

1. Run H on $\langle M, \langle M\rangle\rangle$.
2. Return the answer of H, i.e. if H accepted, then D_1 accepts, if H rejected, then D_1 rejects."

We see that D_1 accepts $\langle D_1\rangle$ if and only if D_1 accepts $\langle D_1\rangle$. This is obviously true and therefore there is no contradiction and there H must exist. Consequently, A_{TM} must be decidable.

Justify your answer.

Exercise 4 (compulsory)

Using the diagonalization method show that the language

$$HALT_{TM} \stackrel{\text{def}}{=} \{\langle M, w\rangle \mid \ M \text{ is a TM and } M \text{ on } w \text{ halts either in accept or reject state}\}$$

is undecidable. Hint: modify slightly the proof of undecidability A_{TM}.

Exercise 5 (optional)

Let us call a Turing machine *repetitive* (abbriviated by RTM), if it *only* loops forever if we encounter the same configuration C more than once during a computation. Is the acceptance problem decidable for the class of repetitive Turing machines? First, define explicitly the language you want to show is decidable/undecidable and then give the arguments.