Tutorial 5

Exercise 1 (compulsory)

Assume that a language A is reducible to language B. Which of the following claims are true?

1. A decider for language A can be used to decide the language B.
2. If A is decidable then B is decidable too.
3. If A is undecidable then B is undecidable too.

Solution:

1. This claim is wrong, the correct claim should be: a decider for language B can be used to decide the language A.
2. This claim is wrong. If A is e.g. the empty language (which is clearly decidable) and B is A_{TM}, then surely $\emptyset$ is reducible to A_{TM}, but A_{TM} is undecidable. The claim is true the other way round: If B is decidable then A is decidable too.
3. This claim is true.

Exercise 2 (compulsory)

1. Give examples of three different languages that are recognizable but not decidable. Argue why the languages are recognizable but not decidable.
2. Give examples of three different languages that are not recognizable. Argue why they are not recognizable.

Solution:

1. $A_{TM}, HALT_{TM}, \overline{E_{TM}}$

   Remark: Note that the languages EQ_{TM} and $REGULAR_{TM}$ neither their complements are recognizable, so this would not be a correct answer.

   - Surely A_{TM} and $HALT_{TM}$ are not decidable (it was proved in Lecture 5). In Lecture 5 we also proved the E_{TM} is undecidable, but this means that $\overline{E_{TM}}$ is undecidable too (if $\overline{E_{TM}}$ was decidable then E_{TM} would be decidable too because decidable languages are closed under complement).
   - A_{TM} and $HALT_{TM}$ are recognizable. The recognizers for these problems would on input $\langle M, w\rangle$ simulate M on w and accept iff the simulation accepted (in case of A_{TM}) or halted (in case of $HALT_{TM}$).

     The argument why $\overline{E_{TM}}$ is recognizable is slightly more difficult. The recognizer would on input $\langle M\rangle$ run in parallel the machine M on all possible inputs (using the dovetailing technique) and accept iff at least one of the parallel computations accepted (and hence the language of M is nonempty). The simulation will of course loop if none of the strings is accepted by M (and hence the language of M is empty).

2. $\overline{A_{TM}}, \overline{HALT_{TM}}, E_{TM}$ (complements of the languages from part 1. of the question)

   Remark: EQ_{TM} and $REGULAR_{TM}$ would be also corrects answers here.

- To show that $\overline{A_{TM}}$, $\overline{HALT_{TM}}$ and E_{TM} are not recognizable recall Theorem 4.22 on page 183 from which we know that:

  If both L and $\overline{L}$ are recognizable then L is decidable.

  As we know that A_{TM}, $HALT_{TM}$, $\overline{E_{TM}}$ are recognizable, their complements cannot be recognizable, because then the languages would be decidable and we know that this is not the case.

Exercise 3 (compulsory)

Consider the following decision problem:

> "Does a given TM M accept a string 0010?"

1. Define this problem as a language L_{0010}.
2. Prove that L_{0010} is undecidable by reduction from A_{TM}.

Solution:

1. $L_{0010} \overset{\text{def}}{=} \{\langle M\rangle \mid \ M \text{ is a TM such that } M \text{ accepts } 0010\ \}$
2. - By contradiction. Assume that there is a decider R for L_{0010}.
   - Using the decider R, we construct a decider S for A_{TM}:

```
S = " On input ⟨M, w⟩:
        1. Using M and w construct the following TM M_1:
              M_1 = " On input x:
                        1. If x ≠ 0010 then M_1 rejects.
                        2. If x = 0010 then simulate M on w.
                           If M accepted then M_1 accepts.
                           If M rejected then M_1 rejects."
        2. Run R on ⟨M_1⟩.
        3. If R accepted then S accepts; If R rejected then S rejects."
```

   - Notice the following properties of the machine M_1: $L(M_1) = \{0010\}$ if M accepts w, and $L(M_1) = \emptyset$ if M does not accept w. Hence $\langle M_1\rangle \in L_{0010}$ if and only if M accepts w, so when R is run on the string $\langle M_1\rangle$ then the machine R will give us the answer to the acceptance problem too.

     This means that if we had the decider R (for L_{0010}), we could construct the decider S (for A_{TM}), but S cannot exist. This implies that R cannot exist either and so L_{0010} is undecidable.

Exercise 4 (compulsory)

Consider the following decision problem:

> "Does a given TM M accept all strings?"

1. Define this problem as a language $TOTAL_{TM}$.
2. Prove that $TOTAL_{TM}$ is undecidable by reduction from A_{TM}.
3. Prove that EQ_{TM} is undecidable by reduction from $TOTAL_{TM}$.

Solution:

1. $TOTAL_{TM} \overset{\text{def}}{=} \{\langle M \rangle \mid M$ is a TM such that $L(M) = \Sigma^* \}$

2.
   - By contradiction. Assume that there is a decider R for $TOTAL_{TM}$.
   - Using the decider R, we construct a decider S for A_{TM}:

     ```
     $S$ = " On input $\langle M, w \rangle$:
         1. Using $M$ and $w$ construct the following TM $M_1$:
             $M_1$ = " On input $x$:
                 1. Ignore the input $x$ and simulate $M$ on $w$.
                    If $M$ accepted then $M_1$ accepts.
                    If $M$ rejected then $M_1$ rejects."
         2. Run $R$ on $\langle M_1 \rangle$.
         3. If $R$ accepted then $S$ accepts; If $R$ rejected then $S$ rejects."
     ```

   - Notice the following properties of the machine M_1: $L(M_1) = \Sigma^*$ if M accepts w, and $L(M_1) = \emptyset$ if M does not accept w. Hence $\langle M_1 \rangle \in TOTAL_{TM}$ if and only if M accepts w, so when R is run on the string $\langle M_1 \rangle$ then the machine R will give us the answer to the acceptance problem too.

     This means that if he had the decider R (for $TOTAL_{TM}$), we could construct the decider S (for A_{TM}), but S cannot exist. This implies that R cannot exist either and so $TOTAL_{TM}$ is undecidable.

3. We now reduce $TOTAL_{TM}$ to EQ_{TM}. By contradiction, assume we have a decider R for EQ_{TM} which on a given input $\langle M_1, M_2 \rangle$ accepts if and only if $L(M_1) = L(M_2)$. We can now use R to construct a decider S for $TOTAL_{TM}$ as follows:

   - for a given input $\langle M \rangle$ of S, we run the decider R on the input $\langle M, M_{total} \rangle$ where M_{total} is a Turing machine that accepts any given string (which means that $L(M_{total}) = \Sigma^*$), and
   - if R accepted then S accepts; if R rejected then S rejects.

   It is easy to see that S is a decider for $TOTAL_{TM}$ but we just showed that such a decider cannot exist, which implies that R cannot exist either and so EQ_{TM} is undecidable.

Exercise 5 (if you feel you need additional practice on reductions)

Problems 5.26 and 5.27 on page 241 (in international edition), or Problems 5.10 and 5.11 on page 239 (in standarad edition).

Solution:
Solutions are given on page 243.

Exercise 6 (optional and slightly challenging)

Problem 5.29 on page 241 (in international edition) or Problem 5.13 on page 239 (in standard edition). (Note the analogy between useless states in Turing machines and dead-code in e.g. java programs.)