Assignment 4 solutions

1. Recall that the language INF is defined as follows:

$$\text{INF} = \{\langle M\rangle \,:\, M \text{ is a DTM and } L(M) \text{ is infinite}\}\,.$$

Prove that there exists a decidable language $B \subseteq \Sigma^*$ such that

$$\text{INF} = \{x \in \Sigma^* \,:\, (\forall y \in \Sigma^*)(\exists z \in \Sigma^*)[\langle x, y, z\rangle \in B]\}\,.$$

Solution. Let M_B be a DTM which on input $\langle\langle M\rangle, y, \langle t\rangle\rangle$, where M is a DTM, $y \in \Sigma^*$ and $t \in \mathbb{N}$, does the following:

1. If $\langle M\rangle$ is not a valid encoding of a DTM, reject.
2. Run M for t steps on each of the t strings following y lexicographically;
3. if M accepts any of them within t steps, then accept; otherwise reject.

The DTM above always stops, therefore the language $B = L(M_B)$ is decidable. The language

$$\{\langle M\rangle \in \Sigma^* \,:\, (\forall y \in \Sigma^*)(\exists t \in \mathbb{N})[\langle\langle M\rangle, y, \langle t\rangle\rangle \in B]\}$$

consists of encodings of all DTMs M such that for every string $y \in \Sigma^*$ there is a string $w > y$ such that M accepts w, which means that this language consists of encodings of all DTMs having infinite languages and, therefore, it is equal to INF.

2. For each string $x \in \Sigma^*$ define a language $B_x \subseteq \Sigma^*$ as

$$B_x = \{\langle M\rangle \,:\, M \text{ is a DTM for which } x \in L(M)\}\,.$$

Prove that $B_x \leq_m B_y$ for every choice of strings $x, y \in \Sigma^*$.

Solution. For $x \in \Sigma^*$ and a DTM M let $K_{M,x}$ be a DTM which on every input $y \in \Sigma^*$ runs M on x. Then $f_x(\langle M\rangle) = \langle K_{M,x}\rangle$ is clearly a computable function. Note that $L(K_{M,x})$ is equal to either $\emptyset$ or Σ^*. Thus, for every $x, y \in \Sigma^*$ we have that $x \in L(M)$ if and only if $y \in L(K_{M,x})$, which implies $B_x \leq_m B_y$.

3. Prove that there does not exist a DTM M with the following property: for every DTM K that halts on all inputs, M accepts $\langle K\rangle$ if and only if $L(K)$ is infinite.

Solution. Suppose the contrary: there exist such a DTM M. We will use this assumption to prove that

$$\text{DIAG} = \{\langle T\rangle \,:\, T \text{ is a DTM and } \langle T\rangle \notin L(T)\}$$

is Turing-recognizable (which we know is false). First, for an arbitrary DTM T, define a DTM K_T as follows:

On input $x \in \Sigma^*$:

1. Run T on $\langle T\rangle$ for $|x|$ steps.

2. If T accepts $\langle T \rangle$ within $|x|$ steps, then reject, otherwise accept.

It holds that K_T always halts, and that $L(K_T)$ is infinite if and only if T does not accept $\langle T \rangle$. The function $f(\langle T \rangle) = \langle K_T \rangle$ is a computable function.

Now consider a DTM D defined as follows:

On input $\langle T \rangle$:

1. Compute $\langle K_T \rangle = f(\langle T \rangle)$.
2. Run M on $\langle K_T \rangle$.

Given that K_T always halts, it holds that M accepts $\langle K_T \rangle$ if and only if $L(K_T)$ is infinite, and therefore D accepts $\langle T \rangle$ if and only if T does not accept $\langle T \rangle$. Therefore $L(D) = \text{DIAG}$, which contradicts the fact that DIAG is not Turing-recognizable.

4. Prove that there exist languages B and C such that these two properties simultaneously hold:
   (a) $B \leq_T C$
   (b) $B \not\leq_m C$ and $B \not\leq_m \overline{C}$.

Solution. Let C be any Turing-recognizable but undecidable language and define

$$B = \{0x : x \in C\} \cup \{1x : x \notin C\}.$$

The following is an oracle Turing machine with an oracle for C which decides B. On input $x \in \Sigma^*$:

1. if $x = \varepsilon$, reject;
2. if $x = 0y$ for $y \in \Sigma^*$, then accept if $y \in C$ or reject if $y \notin C$;
3. we have $x = 1y$ for $y \in \Sigma^*$; accept if $y \notin C$ or reject if $y \in C$.

Therefore $B \leq_T C$.

Suppose $B \leq_m C$. Then, because C is Turing-recognizable, B must be Turing-recognizable as well. However, the solution of Question 2(a) of Midterm Exam 2 states that for an undecidable language C the language $\{0x : x \in C\} \cup \{1x : x \notin C\}$ is not Turing-recognizable. Therefore we have obtained a contradiction, and $B \not\leq_m C$.

Now suppose $B \leq_m \overline{C}$, which implies $\overline{B} \leq_m C$. From the solution of Question 2(b) of Midterm Exam 2 we know that for this particular language B we have $B \leq_m \overline{B}$. Thus, due to transitivity of mapping reductions, $B \leq_m C$, which we already have shown not to be true. Hence, $B \not\leq_m \overline{C}$.

5. Let $f : \mathbb{N} \to \mathbb{N}\backslash\{0\}$ be an arbitrary function, and let $B, C \subseteq \Sigma^*$ be languages such that the symmetric difference $B \triangle C$ is finite. Prove the following logical equivalences:

$$\begin{aligned} B \in \text{DTIME}(f) &\Leftrightarrow C \in \text{DTIME}(f), \\ B \in \text{NTIME}(f) &\Leftrightarrow C \in \text{NTIME}(f), \\ B \in \text{DSPACE}(f) &\Leftrightarrow C \in \text{DSPACE}(f), \\ B \in \text{NSPACE}(f) &\Leftrightarrow C \in \text{NSPACE}(f). \end{aligned}$$

   Hint: once you have the right idea, all of the equivalences will follow—you should not have to consider them separately.

Solution. Let M_B be a DTM deciding B. Then following DTM M_C, which uses M_B as a subroutine. On input $x \in \Sigma^*$:

1. run M_B on x to decide whether $x \in B$, and set $a_1 = 1$ if $x \in B$, or $a_1 = 0$ otherwise;
2. for every $y \in B \triangle C$ check if $x = y$, and set $a_2 = 1$ is such y exists, or $a_2 = 0$ otherwise;
3. if $a_1 \neq a_2$, accept; otherwise reject.

The DTM M_C can be easily seen to decide the language C. Step 3 of M_C takes a constant number of steps, and, because $B \triangle C$ is finite, Step 2 also takes a constant number of steps (note: in order to tell whether $x = y$ we do not need to read more than $|y| + 1$ bits of x). Finally, to simulate M_B running on x we need only linearly more time and space than M_B would use itself. Therefore $B \in$ DTIME$(f) \Rightarrow C \in$ DTIME(f) and $B \in$ DSPACE$(f) \Rightarrow C \in$ DSPACE(f). Due to a symmetry between B and C we also get $B \in$ DTIME$(f) \Leftarrow C \in$ DTIME(f) and $B \in$ DSPACE$(f) \Leftarrow C \in$ DSPACE(f). The analogous argument holds for nondeterministic Turing machines.