Exercise VI, Theory of Computation 2025

These exercises are for your own benefit. Feel free to collaborate and share your answers with other students. Solve as many problems as you can and ask for help if you get stuck for too long. Problems marked * are more difficult but also more fun :).

These problems are taken from various sources at EPFL and on the Internet, too numerous to cite individually.

1 Let A_1 and A_2 be languages with $A_1 \leq_m A_2$. Is it necessarily true that $\overline{A_1} \leq_m \overline{A_2}$?

Solution: Yes, this is true. If $A_1 \leq_m A_2$, then by definition there is a computable function $f : \Sigma^* \to \Sigma^*$ such that for every $w \in \Sigma^*$ we have

$$w \in A_1 \iff f(w) \in A_2.$$

Hence clearly, we also have that for every $w \in \Sigma^*$

$$w \notin A_1 \iff f(w) \notin A_2 \ .$$

This is equivalent to

$$w \in \overline{A_1} \iff f(w) \in \overline{A_2}.$$

Therefore, by definition of the reduction, we have $\overline{A_1} \leq_m \overline{A_2}$ as desired.

2 Classify each of the following languages into one of the following three categories:

decidable, undecidable but recognizable, unrecognizable

Justify your answers with proofs (try reductions).

2a $L_1 = \{\langle M \rangle : M \text{ is a Turing machine that halts on all inputs of length at most 2025}\}$

2b* $L_2 = \{\langle M \rangle : M \text{ is a Turing machine that halts on all inputs of length at least 2025}\}$

2c $L_3 = \{\langle M \rangle : M \text{ is a Turing machine that accepts some string with more zeros than ones}\}$

Solution:

2a The language L_1 is recognizable but not decidable. We show recognizability and undecidability separately. We first construct a Turing machine that recognizes L_1:

On input $\langle M \rangle$, do:

1. Simultaneously, for each string w of length at most 2025, simulate $M(w)$.
2. If ever all simulations halted, accept.

If $\langle M \rangle \in L_1$, then this algorithm accepts since there are finitely many strings of length at most 2025, each accepted by M after finitely many steps. If $\langle M \rangle \notin L_1$, then this algorithm does clearly not accept. Thus, by the Church-Turing thesis, L_1 is recognisable.

To show undecidability, we prove $A_{\text{TM}} \leq_m L_1$. Consider $f : \Sigma^* \to \Sigma^*$ defined as follows:

On input $\langle T, w \rangle$, do:

1. Construct a Turing machine M, which does the following on input x:
   (a) If $T(w)$ accepts, then so does M.
   (b) Otherwise, M enters an infinite loop.
2. Output $\langle M \rangle$.

This algorithm always halts and thus, by the Church-Turing thesis, defines a computable function f. We now show that

$$\langle T, w \rangle \in A_{\text{TM}} \iff f(\langle T, w \rangle) \in L_1.$$

- If $\langle T, w \rangle \in A_{\text{TM}}$, then $T(w)$ accepts and the machine M constructed by f accepts all inputs. In particular, M halts on all inputs of length at most 2025. Thus, $f(\langle T, w \rangle) \in L_1$.
- If $\langle T, w \rangle \notin A_{\text{TM}}$, then the machine M constructed by f loops forever, on all inputs. Thus, M does not halt on inputs of length at most 2025 and $f(\langle T, w \rangle) \notin L_1$.

Therefore, we have that $A_{\text{TM}} \leq_m L_1$. Since A_{TM} is undecidable, we can conclude that L_1 is undecidable as well.

2b The language L_2 is unrecognizable. We show that $\overline{A_{\text{TM}}} \leq_m L_2$, by constructing a computable $f : \Sigma^* \to \Sigma^*$ as follows.

On input $\langle T, w \rangle$, do:

1. Construct a Turing machine M, which does the following on input x:
   (a) If $T(w)$ accepts within $|x|$ steps, enter an infinite loop.
   (b) Otherwise, accept.
2. Output $\langle M \rangle$.

This algorithm clearly halts on all inputs and thus, by the Church-Turing thesis, defines a computable function f. We now show that $\langle T, w \rangle \in \overline{A_{\text{TM}}} \iff f(\langle T, w \rangle) \in L_2$.

- If $\langle T, w \rangle \in \overline{A_{\text{TM}}}$, then $T(w)$ does not accept and the machine M constructed by f accepts all strings. Hence, M halts on all inputs, in particular, all inputs of length at least 2025. We have $\langle M \rangle \in L_2$.
- If $\langle T, w \rangle \notin \overline{A_{\text{TM}}}$, then $T(w)$ accepts within k steps for some $k \in \mathbb{N}$. Hence, the machine M constructed by f loops on all inputs of length more than k, which implies that $\langle M \rangle \notin L_2$.

Therefore, we have that $\overline{A_{\text{TM}}} \leq_m L_2$. Since $\overline{A_{\text{TM}}}$ is unrecognizable, we can conclude that L_2 is unrecognizable as well.

2c The language L_3 is recognizable but not decidable. To show recognizability, we construct a Turing machine that recognizes L_3.

On input $\langle M \rangle$, do:

1. For every $n \in \mathbb{N}$:
   (a) For every $w \in \{0,1\}^*$ such that $|w| \leq n$
      i. If w contains more zeros than ones, simulate $M(w)$ for n steps.
      ii. If this computation accepted, then accept.

Consider some $\langle M \rangle \in L_3$. Then, there exists some string w with more zeros than ones that is accepted by M. Therefore, the above algorithm will eventually find some n such that $|w| \leq n$ and $M(w)$ accepts. Conversely, if $\langle M \rangle \notin L_3$ the above algorithm clearly does not halt. Thus, by the Church-Turing thesis, L_3 is recognisable..

We show the undecidability of L_3 by showing $A_{\text{TM}} \leq_m L_3$. We construct $f : \Sigma^* \to \Sigma^*$ as follows.

On input $\langle T, w \rangle$, do:

1. Construct a Turing machine M that simulates $T(w)$.
2. Output $\langle M \rangle$.

This algorithm always halts and thus defines a computable f. We now show that $\langle T, w \rangle \in A_{\text{TM}} \iff f(\langle T, w \rangle) \in L_3$.

- If $\langle T, w \rangle \in A_{\text{TM}}$, then $T(w)$ accepts, and the machine M given by f accepts on any input. Hence, $\langle M \rangle \in L_3$ as desired.
- If $\langle T, w \rangle \notin A_{\text{TM}}$, then $T(w)$ rejects, and the machine M given by f rejects all inputs. Hence, $\langle M \rangle \notin L_3$ as desired.

Therefore, $A_{\text{TM}} \leq_m L_3$. Since A_{TM} is undecidable, L_3 is undecidable as well.

3 Let A and B be languages. Prove that if all of $A \cap B, \overline{A}$ and $\overline{B}$ are Turing recognisable, then $A \cap B$ is Turing decidable.

Solution: Turing decidable languages are closed under unions, since we can run the two recognisers in parallel and accept as soon as one of them accepts. Thus,

$$\overline{A} \cup \overline{B} = \overline{A \cap B}$$

is also recognisable. But if a language and its complement are both recognisable, then the language is decidable. We conclude that $A \cap B$ is decidable.

4* Consider the language

$$L = \Big\{\langle M\rangle : M \text{ is a Turing machine and } \exists k \in \mathbb{N}\colon M \text{ accepts every string of length} \geq k\Big\}.$$

Prove that L is unrecognizable.

Solution: We show that $\overline{A_{\text{TM}}} \leq_m L$. We construct f as follows.

On input $\langle T, w\rangle$, do:

1. Construct a Turing machine M that does the following on input x:
   (a) Simulate $T(w)$ for $|x|$ many steps.
   (b) If the computation accepted, enter an infinite loop.
   (c) Otherwise, accept.
2. Output $\langle M\rangle$.

This algorithm always halts. We show that $\langle T, w\rangle \in \overline{A_{\text{TM}}} \iff f(\langle T, w\rangle) \in L$.

- If $\langle T, w\rangle \in \overline{A_{\text{TM}}}$, then $T(w)$ rejects, and the machine M given by f accepts all input strings. Hence, $\langle M\rangle \in L$.
- If $\langle T, w\rangle \notin \overline{A_{\text{TM}}}$, then $T(w)$ accepts within l steps for some $l \in \mathbb{N}$. Therefore, the machine M given by f loops forever on every input of length larger than l. We have that $\langle M\rangle \notin L$.

We can conclude that $\overline{A_{\text{TM}}} \leq_m L$. Since $\overline{A_{\text{TM}}}$ is unrecognizable, L is unrecognizable.