CSE 105 Homework 7 Due: Monday December 4, 2017

---

INSTRUCTIONS

Upload a **single file** to Gradescope for each group. All group members' names and PIDs should be on **each** page of the submission.

Your assignments in this class will be evaluated not only on the correctness of your answers, but on your ability to present your ideas clearly and logically. You should always explain how you arrived at your conclusions, using mathematically sound reasoning. Whether you use formal proof techniques or write a more informal argument for why something is true, your answers should always be well-supported. Your goal should be to convince the reader that your results and methods are sound.

READING Sipser Sections 4.1, 4.2, 5.1

KEY CONCEPTS Turing machines, recognizable languages, decidable languages, undecidability

---

1. (**10 points**) **True/False.** Briefly justify your answer for each statement.
   (a) Any subset of a decidable set is decidable.
   **False.** $\Sigma^*$ is decidable, however $A_{TM}$ is not decidable.
   (b) Any subset of a recognizable set is recognizable.
   **False.** $\Sigma^*$ is recognizable, however the complement of $A_{TM}$ is not recognizable.
   (c) There is a decidable but not recognizable language.
   **False.** For any language, if it is decidable, then it is also recognizable.
   (d) There is a recognizable but not decidable language.
   **True.** $A_{TM}$ is recognizable but not decidable.
   (e) Recognizable sets are closed under complement.
   **False.** $A_{TM}$ is recognizable but its complement is not recognizable.
   (f) Decidable sets are closed under complement.
   **True.** We just need to flip the accept states and reject states.
   (g) Recognizable sets are closed under intersection.
   **True.** By constructing Turing machine.
   (h) Decidable sets are closed under intersection.
   **True.** By constructing Turing machine.
   (i) Every language reduces to itself.
   **True.** By trivial reduction.
   (j) Every language reduces to its complement.
   **False.** $A_{TM}$ can not reduce to its complement.

2. (**10 points**) Determine whether the following languages are decidable, recognizable, or undecidable. Briefly justify your answer for each statement.
   (a) $L_1 = \{\langle D, w\rangle : D \text{ is a DFA and } w \notin L(D)\}$
   **Decidable.** From the lecture, we know the complement of $L_1$ is decidable, so does $L_1$.
   (b) $L_2 = \{\langle N, w\rangle : N \text{ is a NFA and } w \in L(N)\}$
   **Decidable.** We can convert a NFA $N$ to a DFA, and hence $L_2$ reduces to $L_1$.

(c) $L_3 = \{\langle P, w\rangle : P$ is a PDA and $w \in L(P)\}$

**Decidable.** Similar to $L_1$.

(d) $L_4 = \{\langle M, w\rangle : M$ is a TM and $w \in L(M)\}$

**Undecidable, but recognizable** We know it from the lecture.

(e) $L_5 = \{\langle M, w\rangle : M$ is a TM and $w \notin L(M)\}$

**Not recognizable.** Since its complement is recognizable but not decidable.

3. (**10 points**) Prove that the language

$$\{\langle D, Q\rangle \mid D \text{ and } Q \text{ are DFAs and } L(D) \cap L(Q) = \emptyset\}$$

is decidable.

**Proof:** We construct the following Turing machine $M$ on input $\langle D, Q\rangle$ to decide the above language.

- Let $D$ and $Q$ be the input DFAs.
- We construct a DFA $P$ that accepts $L(D) \cap L(Q)$ using the Cartesian product construction proved in class.
- Run the Turing machine $T$ that decides $E_{DFA}$ on input $\langle P\rangle$ (see Sipser, Theorem 4.4).
- If $T$ accepts then **accept**; if $T$ rejects then **reject**.

Now $M$ accepts $\langle D, Q\rangle$ if and only if $E_{DFA}$ accepts $\langle P\rangle$ e.g., if and only if $L(P) = \emptyset$. By the construction of $P$, we have that $L(P) = L(D) \cap L(Q)$, the correctness then follows since $M$ accepts $\langle D, Q\rangle$ if and only if $L(D) \cap L(Q) = \emptyset$.

4. (**10 points**) Prove that the language

$$L = \{\langle M\rangle : M \text{ is a TM and } M \text{ accepts the empty string}\},$$

is undecidable.

**Proof:** We prove it by contradiction. We first assume that $L$ is decidable, say the TM $M_L$ decides $L$. We prove that $A_{TM}$ is decidable. Remind that

$$A_{TM} := \{\langle M, x\rangle : M \text{ accepts } x\}$$

For each pair of $M$ and $x$, we define the Turing machine $M_x$ as

- $M_x$. On input $w$,
- Erase $w$ and replace it by $x$.
- Run $M$ on the input $x$.

Then we construct the Turing machine $M_{ATM}$ as follows,

- On input $\langle M, x\rangle$.
- Construct the Turing machine $\langle M_x\rangle$.
- Run $M_L$ on the input $\langle M_x\rangle$.
- If $M_L$ accepts then **accept**; if $M_L$ rejects then **reject**.

Correctness: We prove correctness by two directions.

- If $\langle M, x\rangle \in A_{TM}$, e,g, $M$ accepts $x$. By the definition of $M_x$, it accepts everything, which means $L(M_x) = \Sigma^*$. In particular, we have that $\epsilon \in L(M_x)$. Since $M_L$ decides the language $L$, $M_L$ must accept $\langle M_x\rangle$, thus by the definition of $M_{ATM}$, it accepts $\langle M, x\rangle$.
- If $\langle M, x\rangle \in A_{TM}$, e,g, $M$ does not accepts $x$. By the definition of $M_x$, it accepts nothing, which means $L(M_x) = \emptyset$. In particular, we have that $\epsilon \notin L(M_x)$. Since $M_L$ decides the language $L$, $M_L$ must reject $\langle M_x\rangle$, thus by the definition of $M_{ATM}$, it rejects $\langle M, x\rangle$.

Combine these two cases, we show that $M_{ATM}$ decides $A_{TM}$, which is a contradiction since we know that $A_{TM}$ is undecidable.

5. (**10 points**) Prove that the language

$$L = \{\langle M_1, M_2\rangle : M_1, M_2 \text{ are TMs and } L(M_1) = L(M_2)\},$$

is undecidable.

**Proof:** We prove it by contradiction. We first assume that $L$ is decidable, say the TM $M_L$ decides $L$. We prove that $A_{TM}$ is decidable. Similar as the previous solution, for each pair of $M$ and $x$, we define the function $M_x$. We also define the Turing machine $M_c$ that accepts every string. Then we construct the Turing machine $M_{ATM}$ as follows,

- On input $\langle M, x\rangle$.
- Construct the Turing machines $\langle M_x, M_c\rangle$.
- Run $M_L$ on the input $\langle M_x, M_c\rangle$.
- If $M_L$ accepts then **accept**; if $M_L$ rejects then **reject**.

Correctness: We prove correctness by two directions.

- If $\langle M, x\rangle \in A_{TM}$, e,g, $M$ accepts $x$. By the definition of $M_x$, it accepts everything, which means $L(M_x) = \Sigma^*$. Thus $L(M_x) = L(M_c)$. Since $M_L$ decides the language $L$, $M_L$ must accept $\langle M_x, M_c\rangle$, thus by the definition of $M_{ATM}$, it accepts $\langle M, x\rangle$.
- If $\langle M, x\rangle \in A_{TM}$, e,g, $M$ does not accepts $x$. By the definition of $M_x$, it accepts nothing, which means $L(M_x) = \emptyset$. Thus $L(M_x) \neq L(M_c)$. Since $M_L$ decides the language $L$, $M_L$ must reject $\langle M_x, M_c\rangle$, thus by the definition of $M_{ATM}$, it rejects $\langle M, x\rangle$.

Combine these two cases, we show that $M_{ATM}$ decides $A_{TM}$, which is a contradiction since we know that $A_{TM}$ is undecidable.