Solutions of Assignment 9

> Theory of Computation, Fall 2022

Keys

Q1

(a)

$$A_w = \{''w'' : D \text{ accepts } w\} \tag{1}$$

i.e.

$$A_w = L(D) \tag{2}$$

(b)

Yes.

(c)

Proof:

For every regular language R, we can construct a DFA $D = (K, \Sigma, \delta, s, F)$ that accepts it.

Construct a Turing machine M that decides R as follows.

$$\begin{aligned} M \quad = \quad & \text{on input } w: \\ & \text{1. run } D \text{ on } w \\ & \text{2. if } D \text{ accepts } w \\ & \text{3.} \quad \text{accept } ''w'' \\ & \text{4. else} \\ & \text{5.} \quad \text{reject } ''w'' \end{aligned} \tag{3}$$

Therefore, every regular language is decided by some Turing machine, hence every regular language is recursive.

Q2

Suppose Turing machine M_E decides E_{DFA}.

To give the reduction from SB_{EDA} to E_{DFA}, we construct a Turing machine M that decides SB_{DFA} as follows.

$$\begin{aligned} M \quad = \quad & \text{on input } "D_1""D_2" : \\ & \text{1. construct a DFA } D^- \text{ with } L(D^-) = L(D_1) \backslash L(D_2) \\ & \text{2. run } M_E \text{ on } "D^-" \\ & \text{3. output the result} \end{aligned} \tag{4}$$

The reduction is $f("D_1""D_2") = "D^-"$ where $L(D^-) = L(D_1) \backslash L(D_2) = L(D_1) \cap \overline{L(D_2)}$.

Q3

Suppose Turing machine M_{EQ} decides EQ_{DFA}.

To give the reduction from A_L to EQ_{DFA}, we construct a Turing machine M that decides A_L as follows.

$$\begin{aligned} M \quad = \quad & \text{on input } "D" : \\ & \text{1. construct a DFA } D_0 \text{ with } L(D_0) = L \\ & \text{2. run } M_{EQ} \text{ on } "D""D_0" \\ & \text{3. output the result} \end{aligned} \tag{5}$$

The reduction is $f("D") = "D""D_0"$ where $L(D_0) = L$.

Q4

SB_{DFA} and A_L are also recursive.

Q5

(a)

A is also recursively enumerable.

Proof:

Since B is recursively enumerable, there is a Turing machine M_B semidecides B, that is, M_B halts on every string $y \in B$ and loops on every string $y' \in \Sigma_B^* \backslash B$.

Construct a M_A based on M_B using reduction f.

Since $x \in A$ if and only if $f(x) \in B$, M_A halts on every string $x \in A$ and loops on every string $x' \in \Sigma_A^* \backslash A$, that is, M_A semidecides A.

Therefore, A is also recursively enumerable.

(b)

B is also NOT recursive.

Proof:

Assume that B is recursive, then there is a Turing machine M_B decides B, that is, M_B accepts every string $y \in B$ and rejects every string $y' \in \Sigma_B^* \backslash B$.

Construct a M_A based on M_B using reduction f.

Since $x \in A$ if and only if $f(x) \in B$, M_A accepts every string $x \in A$ and rejects on every string $x' \in \Sigma_A^* \backslash A$, that is, M_A decides A.

However, A is NOT recursive, which means there is NO Turing machine decides A, leading to contradiction.

Therefore, the assumption fails, that is, B is also NOT recursive.

Q6

Proof:

- Lemma 1: Σ^* is countable.
- Lemma 2: A subset of a countable set is countable.
- Corollary: A subset of Σ^* is countable.
- Lemma 3: Every language is a subset of Σ^*.
- Conclusion: Every language is countable.

Q7

Proof:

- Lemma 1: The set of all Turing machines is countable.
- Lemma 2: Every Turing machine decides at most 1 language.
- Corollary: The set of all recursive languages is countable.
- Lemma 3: The power set of $\{1\}^*$ is uncountable.
- Conclusion: There is an undecidable subset of $\{1\}^*$.

Grading

100 pts. in total

- Q1. 22 pts. = (a) 5 pts. + (b) 2 pts. + (c) 15 pts.
- Q2. 12 pts.
- Q3. 12 pts.
- Q4. 4 pts.
- Q5. 30 pts. = (a) 15 pts. + (b) 15 pts.
- Q6. 10 pts.
- Q7. 10 pts.
