Sample solution to HW 1

Question 1.a.

```
q0 --b--> q1 --a--> q2
```

Question 1.b.

```
a,b
q1 -<a--> q0 -<b--> q2
```

Question 1.c. Intuitively, we can construct a DFA that as soon as it “sees” b, it cannot see any a. This is formalized by the following DFA.

```
q0 -<a--> q2 -<b--> q1 --b--> q2
```

Question 1.d. Define the following two languages.

\[ K_1 = \{ w \mid w \text{ does not contain } ba \} \]
\[ K_2 = \{ w \mid w \text{ contains } ba \text{ and ends with } bb \} \]

Note that \( L_4 = K_1 \cup K_2 \), and \( K_1 \) is the language in question 1.c. This observation already gives us a hint on how to construct an NFA for \( L \) by modifying the DFA for \( K_1 \).

Notice that in the DFA for question 1.c, we can reach state \( q_2 \), if the word contains \( ba \). Therefore, we have to make sure that it ends with \( bb \) after state \( q_2 \), by adding two more states \( q_3 \) and \( q_4 \) to ensure that \( bb \) appears at the end word. This is formalized by the following NFA.

```
q0 -<a--> q1 -<b--> q2 -<a,b--> q3 -<b--> q4
```

Question 2.a. \( b\Sigma^*a \).

Question 2.b. \( \Sigma^*aba\Sigma^* \).

Question 2.c. There are two ways to construct the regex for this language. One is by constructing the regex from the DFA in question (1.d) via the procedure described in the proof of Theorem 4.1. Another way is to observe the following: \( w \) does not contain \( ab \), if and only if one of the following holds.

\[ w \text{ does not contain } ab \]

\[ w = z_1z_2 \]

\[ z_2 = z_3z_4 \]

\[ z_3 \text{ does not contain } ab \]

\[ z_4 = \epsilon \]
• $w$ does not contain any $b$, i.e., it contains only $a$’s.
• Once a $b$ appears, no $a$ will appear.

With this observation, we can immediately construct the desired regex $e_3$ for $L_3$:

$$e_3 := a^* \cup a^*bb^*.$$

**Question 2.d.** Recall that $L_4 = K_1 \cup K_2$, as mentioned in question (1). The regex for $K_1$ is as in question (2.c). The regex for $K_2$ is $e := \Sigma^*ba\Sigma^*bb$. So the desired regex for $L_4$ is:

$$e_4 = a^* \cup a^*bb^* \cup \Sigma^*ba\Sigma^*bb.$$

**Question 3.a.** $L_5$ is regular with regex: $b^*ab^*$.  

**Question 3.b.** $L_6$ is regular with regex: $b^* \cup (b^*ab^*)^*$.  

**Question 3.c.** $L_7$ is not regular. The proof is via pumping lemma. Suppose to the contrary that $L_7$ is regular. Let $A = (\Sigma, Q, q_0, F, \delta)$ be its NFA.

Consider the word $a^m ba^i$, where $m > n \geq |Q|$. By Lemma 3.6, i.e., more refined pumping lemma, we can partition $a^n$ into three parts $uvw$ such that for each $i \geq 0$, $a^m bu v^i w$ is accepted by $A$, which is not possible. We can choose an $i \geq m$ which makes the number of $a$’s on the right hand side of $b$ bigger than $m$. Thus, there is no such NFA that accepts $L$.

**Question 3.d.** $L_8$ is not regular. The proof is again via pumping lemma (Lemma 3.6). Suppose to the contrary that $L$ is regular. Let $A = (\Sigma, Q, q_0, F, \delta)$ be its NFA. Without loss of generality, we can assume that $|Q| \geq 2$. (We can always add some redundant states.)

Consider the word $a^m a^{|Q|}$, where $m + |Q|$ is a prime number and $m \geq |Q|$. By Lemma 3.6 we can partition $a^m$ into three parts $uvw$ such that for each $i \geq 0$, $u v^i w a^{|Q|}$ is accepted by $A$.

Now, $u v^i w a^{|Q|}$ consists of only $a$’s. So, the number of $a$’s in it is precisely its length, which is:

$$|u v^i w a^{|Q|}| = |u| + i|v| + |w| + |Q|$$

If we consider $i = |u| + |w| + |Q|$, we have:

$$|u v^i w a^{|Q|}| = |u| + (|u| + |w| + |Q|)|v| + |w| + |Q|$$

$$= (|u| + |w| + |Q|)(|v| + 1)$$

Thus, $|u v^i w a^{|Q|}|$ cannot be a prime number. However, it is supposed to be accepted by $A$. Therefore, there cannot be such NFA $A$ that accepts $L$.

**Question 4.** For a language $L \subseteq \Sigma^*$ (not necessarily regular), we define the equivalence relation $\sim_L$ on $\Sigma^*$ as follows. $u \sim_L v$, if the following holds: For every $w \in \Sigma^*$, $uw \in L$ if and only if $vw \in L$. Prove that $\sim_L$ is an equivalence relation and that $L$ is a regular language if and only if $\#(\sim_L)$ is finite.\[*\]

That $\sim_L$ is an equivalence relation is quite straightforward. So, we present the proof of the second part. We start with the “only if” part. Let $L$ be a regular language and $A$ be its DFA.

For a word $w$, we denote by $A(w)$ the state of $A$ after reading $w$. Or, more formally, if $w = a_1 \cdots a_n$ and $q_0 a_1 q_1 \cdots a_n q_n$ is the run of $A$ on $w$, then $A(w) = q_n$.

We will first prove the following:

\[*\] This theorem is usually called Myhill-Nerode theorem.
Claim 1 \textit{For every words }u, v, \textit{if }A(u) = A(v), \textit{then }u \sim_L v.\]

\textbf{Proof.} Let }u\text{ and }v\text{ be such that }A(u) = A(v). \text{ Let } u = a_1 \cdots a_n \text{ and } v = b_1 \cdots b_m. \text{ We have to show that for every } w \in \Sigma^*, uw \in L \text{ if and only if } vw \in L. \text{ Let } w = c_1 \cdots c_k. \text{ Consider the run of } A \text{ on } uw:
\begin{align*}
p_0 a_1 p_1 \cdots \cdots a_n p_n c_1 r_1 \cdots \cdots c_k r_k
\end{align*}
Likewise, consider the run of } A \text{ on } vw:
\begin{align*}
s_0 b_1 s_1 \cdots \cdots b_m s_m c_1 t_1 \cdots \cdots c_k t_k
\end{align*}
Here both }p_0, s_0\text{ is the initial state of } A. \text{ Since } A(u) = A(v), \text{ we have } p_n = s_m. \text{ Furthermore, } A \text{ is deterministic. Thus, } r_1 = t_1, \ldots, r_k = t_k, \text{ and therefore,}
\begin{align*}
A(uw) &= A(vw)
\end{align*}
This completes proof of Claim 1. \hfill \blacksquare

Claim 1 immediately implies that \(#(\sim_L) \leq |Q|\), where }Q\text{ is the set of states of } A. \text{ Thus,}
\begin{align*}
\#(\sim_L) \text{ is finite.}
\end{align*}
Now, we show the “if” direction. \text{ Let } L \text{ be a language over } \Sigma, \text{ where } \sim_L \text{ has finitely many equivalence classes } C_1, \ldots, C_m. \text{ Without loss of generality, we can assume that } L \neq \emptyset. \text{ We first prove the following claim.}

Claim 2 \textit{There is } i_1, \ldots, i_k \subseteq \{1, \ldots, m\} \text{ such that } L = C_{i_1} \cup \cdots \cup C_{i_k}. \text{ In other words, } L \text{ is a union of some of the equivalence classes of } \sim_L.\]

\textbf{Proof.} \text{ Note that if } w \sim_L v, \text{ then either both of them belong to } L, \text{ or both of them do not belong to } L. \text{ Thus, Claim 2 follows immediately.} \hfill \blacksquare

Now, consider the following DFA } A = (\Sigma, Q, q_0, F, \delta).
\begin{itemize}
\item \text{ } Q = \{p_1, \ldots, p_m\}, \text{ i.e., the number of states is precisely the number of equivalence classes in } \sim_L.
\item \text{ } q_0 \text{ is } p_j, \text{ where } j \text{ is such that } \epsilon \in C_j.
\item \text{ } F = \{p_{i_1}, \ldots, p_{i_k}\}, \text{ where } i_1, \ldots, i_k \text{ are the indices in (5.a).}
\item \text{ } \delta : Q \times \Sigma \rightarrow Q \text{ is defined as follows. For every } p_i \in Q, \text{ for every } a \in \Sigma, \text{ we pick an arbitrary } w \in C_i, \text{ and define } \delta(p_i, a) = p_j, \text{ where } [wa] = C_j.
\end{itemize}
Note that } \delta \text{ is a well-defined function, i.e., for every } w_1, w_2 \in C_i, [w_1 a] = [w_2 a]. \text{ In other words, the end result } p_j \text{ remains the same for whichever } w \text{ we pick, as long as } w \text{ is from } C_i.

\text{ We will show that } L(A) = L. \text{ Recall that } A(w) \text{ is the state of } A \text{ after reading } w \text{ starting from the initial state. From the construction of } A, \text{ for every word } w \in \Sigma^*, \text{ if } [w] = C_j, \text{ then } A(w) = p_j. \text{ Now,}
\begin{align*}
w \in L &\text{ if and only if } w \in C_{i_1} \cup \cdots \cup C_{i_k}
\end{align*}
and hence,
\begin{align*}
w \in C_{i_1} \cup \cdots \cup C_{i_k} \text{ if and only if } A(w) \text{ is one of } p_{i_1}, \ldots, p_{i_k}.
\end{align*}
Thus, } w \in L \text{ if and only if } w \in L(A), \text{ and hence, } L = L(A).

\text{1Here, for } w \in \Sigma^*, [w] \text{ denotes the equivalence class of } \sim_L \text{ that contains } w. \text{ See the notation in Lecture 1.}
Question 5. In the following, we simply write $\sim$ to denote $\sim_{L_9}$. First, note that if $m \neq n$, we have: $ab^m \not\sim ab^n$. This is because $ab^m ab^m \in L_9$ but $ab^m ab^n \notin L_9$ (due to the fact that $m \neq n$).

Therefore, we have infinitely many words $ab, ab^2, ab^3, \ldots$ where each of them belongs to different equivalent classes of $\sim$. Thus, the index $\#(\sim)$ is infinite, and hence, $L_9$ is not regular.

---

\[ \text{We write } u \not\sim v \text{ to denote that “it is not true that } u \sim v. \]