Theory of Computation

Spring 2025, Homework # 2 (Solutions)

1. (15 pts) Let $L_1 \subseteq (a+b)^*$ be a set of strings. In each string in L_1, delete every b immediately preceding (i.e., before) an a to get the set L_2. For instance, if $L_1 = \{aabba, aa\}$, then $L_2 = \{aaba, aa\}$. You are asked to define two homomorphisms $h_1, h_2 : \{a, b, \hat{b}\}^* \to \{a, b\}^*$), and write an expression for L_2 in terms of $h_1, h_2, h_1^{-1}, h_2^{-1}, R, L_1$, for some regular expression R. Explain why your answer is correct.
Answer.
Let R be the regular expression of laguage

$$\{\text{Every } \hat{b} \text{ immediately following an } a, \text{and no b immediately preceding an a}\}.$$

Define

$$h_1 : \{a, b, \hat{b}\}^* \to \{a, b\}^*, h_1(a) = a, h_1(b) = b, h_1(\hat{b}) = b, h_1(w_1w_2) = h_1(w_1)h_1(w_2)$$

$$h_2 : \{a, b, \hat{b}\}^* \to \{a, b\}^*, h_2(a) = a, h_2(b) = b, h_2(\hat{b}) = \varepsilon, h_2(w_1w_2) = h_2(w_1)h_2(w_2).$$

Then $h_1^{-1}(L_1) = \{x \mid h_1(x) \in L_1\}$,
and $h_1^{-1}(L_1) \bigcap R = \{x \mid b$'s immediately preceding an a are replaced by $\hat{b}\}$,
and $L_2 = h_2(h_1^{-1}(L_1) \bigcap R)$.
In the example $L_1 = \{aabba, aa\}$, $h_1^{-1}(L_1) = \{aabba, aab\hat{b}a, aa\hat{b}ba, aa\hat{b}\hat{b}a, aa\}$.$h_1^{-1}(L_1) \bigcap R = \{aab\hat{b}a, aa\}$.$h_2(h_1^{-1}(L_1) \bigcap R) = \{aaba, aa\}$.

2. (15 pts) For a language L, let $Init(L) = \{x \mid xy \in L, \text{ for some } y \in \Sigma^*\}$. Let r, s, r_I and s_I be regular expressions for the languages R, S, $Init(R)$, and $Init(S)$, respectively. Using only these regular expressions and the operations +, concatenation, and *, give expressions for the following languages. Briefly explain why your answers work.

   (a) $Init(R \cup S)$
   (b) $Init(RS)$
   (c) $Init(R^*)$

   Answer.
   (a) $r_I + s_I$

$$\begin{aligned} Init(R \cup S) &= \{x \mid xy \in R \cup S, \text{for some } y \in \Sigma^*\} \\ &= \{x \mid xy \in R, \text{for some } y \in \Sigma^*\} \cup \{x \mid xy \in S, \text{for some } y \in \Sigma^*\} \\ &= Init(R) + Init(S) \end{aligned}$$

   (b) $r_I + rs_I$

$$\begin{aligned} Init(RS) &= \{x \mid xy \in RS, \text{for some } y \in \Sigma^*\} \\ &= \{x \mid xy = rs, \text{for some } y \in \Sigma^*, r \in R, s \in S\} \\ &= Init(R) + RInit(S) \end{aligned}$$

   Note that if $|x| \leq |r|$ then $x \in Init(R)$, else $x \in RInit(S)$.
   (c)r^*r_I

$$\begin{aligned} Init(R^*) &= \{x \mid xy \in R^*, \text{for some } y \in \Sigma^*\} \\ &= \{x \mid xy \in \bigcup_{i=0}^{\infty} R^i, \text{for some } y \in \Sigma^*\} \\ &= \bigcup_{i=0}^{\infty} \{x \mid xy \in R^i, \text{for some } y \in \Sigma^*\} \\ &= R^*Init(R) \end{aligned}$$

   Note that we know $Init(RR) = Init(R) + RInit(R)$ in (b).

3. (30 pts) Given a language L, we define (as discussed in class) the relation $\equiv_L$ for strings x and y in Σ^* as

$$x \equiv_L y \Leftrightarrow (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L).$$

Suppose we consider the following relations $\equiv^L$ and $\overset{L}{\equiv}$ instead

$$x \equiv^L y \Leftrightarrow (\forall z \in \Sigma^*, zx \in L \Leftrightarrow zy \in L).$$

$$x \overset{L}{\equiv} y \Leftrightarrow (\forall u, v \in \Sigma^*, uxv \in L \Leftrightarrow uyv \in L).$$

Given a string w, let w^R be its reverse, i.e., the word obtained when reading w from right to left, e.g., $(abb)^R = bba$. For a language L, we let $L^R = \{w^R \mid w \in L\}$.

(a) (5 pts) Prove in detail that $(x \equiv_L y) \Leftrightarrow (x^R \equiv^{L^R} y^R)$.

(b) (10 pts) Consider langauge $L = (ab + ba)^*$. Determine the equivalence classes of the language $(ab + ba)^*$ under $\equiv_L$. Write down each of the equivalence classes using a regular expression.

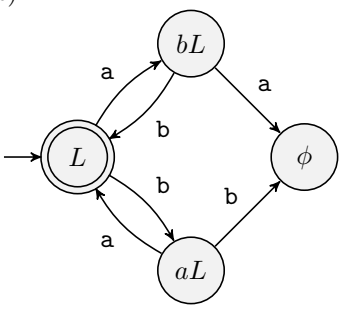
(c) (10 pts) Use the above equivalence classes to construct (draw) a DFA accepting $(ab+ba)^*$.

(d) (5 pts) For $L = (ab + ba)^*$, can you find two strings x and y such that both $x \equiv_L y$ and $x \equiv^L y$ hold, but $x \overset{L}{\equiv} y$ does not hold. Justify your answer.

Answer.
(a)

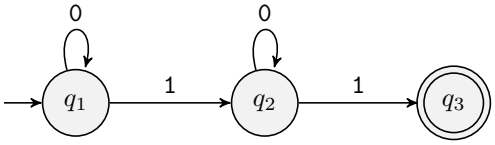
$$\begin{aligned} x \equiv_L y &\Leftrightarrow (\forall z \in \Sigma^*, xz \in L \Leftrightarrow yz \in L). \\ &\Leftrightarrow (\text{Given any } z = z_1...z_k, \text{ we have } x_1...x_{|x|}z_1...z_k \in L \Leftrightarrow y_1...y_{|y|}z_1...z_k \in L). \\ &\Leftrightarrow (\text{Given any } z^R = z_1...z_k, \text{ we have } z_k...z_1x_{|x|}...x_1 \in L^R \Leftrightarrow y_1...y_{|y|}z_1...z_k \in L^R). \\ &\Leftrightarrow (\forall z \in \Sigma^*, zx^R \in L^R \Leftrightarrow zy^R \in L^R). \\ &\Leftrightarrow x^R \equiv^{L^R} y^R \end{aligned}$$

(b)$\{L, bL, aL, \phi\}$
(c)



(d) Let $x = \varepsilon, y = ab$. It's clear that $x \equiv_L y$ and $x \equiv^L y$ hold.
Choose $u = a, v = b$, we have $uxv \in L$, and $uyv = aabb \notin L$.

4. (20 pts) Construct a DFA for $\Sigma^*1(\Sigma\Sigma^*)^*1\Sigma^*$ with the smallest possible number of states, where $\Sigma = \{0, 1\}$. Prove that your DFA is the smallest possible. (Hint: To prove the DFA to be minimum, you may use the Myhill-Nerode theorem.)
Answer.

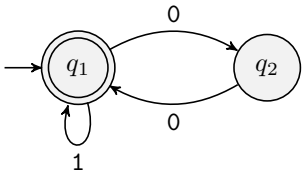


$\Sigma^*1(\Sigma\Sigma^*)^*1\Sigma^* = \Sigma^*1\Sigma^*1\Sigma^* = \{$ strings contain 2 1's $\}$, so there are 3 equivalence classes: one is {strings contain 2 1's}, other one is {strings contain only one 1}, the last one is {strings contain no 1}.

5. (10 pts) Let B and D be two languages. We write $B \propto D$ if $B \subseteq D$ and D contains infinitely many strings that are not in B. Show that, if B and D are two regular languages where $B \propto D$, then we can find a regular language C such where $B \propto C \propto D$.
   Answer.
   Let $A = D - B = D \cap \overline{B}$, A is regular and not a finite set.
   So we can find a string $s \in A$ s.t. $|s|$ is larger than the pumping length p of language A. Choose the partition $s = xyz$ in pumping lemma.
   Let $C = \{xy^{2i}z \mid i \geq 0\} \cup B$. It is obvious that $B \propto C$ according to pumping lemma. — (1)
   We know $C \subseteq D$, and $\{xy^{2i+1}z \mid i \geq 0\} \subseteq D - C$ which imply $C \propto D$ — (2)
   $(1)(2) \implies B \propto C \propto D$

6. (10 pts) Give a right-linear grammar for the following regular language: $(00 \cup 1)^*$. Show your work in sufficient detail.

Answer. Below is a DFA M decide $(00 \cup 1)^*$.



we can construct a right-linear grammar G_M such that the language of G is $L(M)$. You can see the example in CH2. pg 8.
Let $G = (\{R_1, R_2\}, \{0, 1\}, \{R_1 \to 0R_2 \mid 1R_1 \mid \varepsilon; \quad R_2 \to 0R_1\}, R_1)$ be the G_M.