CSC 544 Homework 1

Calvin Higgins

September 2025

Problem 1

Let $Z = \{01^k0 \mid k \geq 0\}$.

Part (a)

Give a GNFA that recognizes Z.

The GNFA has two states: one is the start state, the other is the accept state. There is a single arrow from the start state to the accept state labeled with the regular expression 01^*0.

Part (b)

Give a regular expression that generates Z.

The expression 01^*0 generates Z.

Problem 2

Part (a)

Let $L = \{\text{abra}(\text{cad})^k\text{abra} \mid k \geq 1\}$. Prove or disprove: L is regular.

Proof. We will show that L is regular by construction. Let $A = \{\text{abra}\}$ and $C = \{\text{cad}\}$. Since A and C have finite cardinality, they are regular. Define $L' = A \circ C \circ C^* \circ A$. Since A and C are regular, and the class of regular languages is closed under $\circ$ and $*$, L' is regular.

We will argue that $L' = L$. Note that

$$\begin{aligned} L' &= A \circ C \circ C^* \circ A \\ &= \{\text{abra}\} \circ \{\text{cad}\} \circ \{(\text{cad})^k \mid k \geq 0\} \circ \{\text{abra}\} \\ &= \{\text{abra}\} \circ \{(\text{cad})^k \mid k \geq 1\} \circ \{\text{abra}\} \\ &= \{\text{abra}(\text{cad})^k\text{abra} \mid k \geq 1\} \\ &= L \end{aligned}$$

so L is regular as desired. //

Part (b)

Let $L = \{(\text{abra})^r(\text{cad})^k(\text{abra})^r \mid k \geq 1, r \geq 0\}$. Prove or disprove: L is regular.

Proof. We will show that L is not regular via the pumping lemma. Assume that L is regular, and let p be the pumping length. Take

$$s = (\text{abra})^p(\text{cad})(\text{abra})^p$$

so $s \in L$.

We will show that s cannot be pumped. For any choice of u, v, w such that $uvw = s$ and $1 \leq |uv| \leq p$, v contains some fragment of some abra to the left of cad. Pumping down, we have that $s' = uv^0w = uw \in L$. However, s' contains too few symbols to form enough abra's on the left to match the p abras on the right (*the casework is too painful for anything but hand waving*), so $s' \notin L$, which is impossible. Hence L is not regular. //

Problem 3

Let $\Sigma = \{0, 1\}$.

Part (a)

Let $L = \{xyx \mid x, y \in \Sigma^*\}$. Prove that L is regular.

Proof. We will show that L is regular by construction. The main idea is that L contains all strings (e.g. take $x = \epsilon$). Since Σ has finite cardinality, Σ is regular. Define $L' = \Sigma^*$. Since Σ is regular, and the class of regular languages is closed under $*$, L' is regular.

We will argue that $L' = L$. Clearly $L \subseteq L' = \Sigma^*$. Fix some $s \in L'$. Let $x = \epsilon$ and $y = s$. Then $x \in \Sigma^*$ and $y \in L' = \Sigma^*$. By definition, $xyx \in L$, so $xyx = \epsilon s \epsilon = s \in L$. Therefore, $L' \subseteq L$. Hence, $L' = L$, so L is regular as desired. //

Part (b)

Let $L = \{xyxy \mid x, y \in \Sigma^*\}$. Prove that L is not regular.

Proof. We will show that L is not regular via the pumping lemma. Assume that L is regular, and let p be the pumping length. Take $x = 0^p 1^p$ and $y = 0^p 1^p$. Then $x, y \in \Sigma^*$, so $s = xyxy \in L$.

We will show that s cannot be pumped. For any choice of u, v, w such that $uvw = s$ and $1 \leq |uv| \leq p$, $v = 0^k$ for some $1 \leq k \leq p$ since the first p symbols in s are all zero. By the pumping lemma, $s' = uv^0w = 0^{p-k}1^p0^p1^p \in L$. Suppose there exists some choice of x', y' such that $s' = x'y'x'y'$. Then, the first half of s' is $x'y' = 0^{p-k}1^p1^{k/2}$ and the second half of s' is $x'y' = 1^{p-k/2}0^p$. However, the first half is not the same as the second half, so $x'y' \neq x'y'$ which is impossible. Thus, there is no choice of $x'y'$ such that $s' = x'y'x'y'$, so $s' \notin L$, which is impossible. Hence L is not regular as desired. //

Problem 4

Part (a)

Prove or disprove: the class of regular languages is closed under union.

Proof. Let L_1 and L_2 be regular languages. We will show that $L = L_1 \cup L_2$ is regular by construction. The main idea is to simulate M_1 and M_2 simultaneously, and accept if either accepts.

Since L_1 is regular, there exists some DFA $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$ that recognizes L_1, and since L_2 is regular, there exists some DFA $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$ that recognizes L_2. Define DFA

$$M = ((Q_1 \cup q_{\text{reject}}) \times (Q_2 \cup q_{\text{reject}}), \Sigma_1 \cup \Sigma_2, \delta, (q_0^1, q_0^2), (F_1 \times Q_2) \cup (Q_1 \times F_2))$$

where

$$\delta(s_k, (q_i^1, q_j^2)) = \begin{cases} (\delta_1(q_i^1), \delta_2(q_j^2)) & \text{if } s_k \in \Sigma_1 \wedge s_k \in \Sigma_2 \wedge q_i^1 \in Q_1 \wedge q_j^2 \in Q_2 \\ (\delta_1(q_i^1), q_{\text{reject}}) & \text{if } s_k \in \Sigma_1 \wedge s_k \notin \Sigma_2 \wedge q_i^1 \in Q_1 \\ (q_{\text{reject}}, \delta_2(q_j^2)) & \text{if } s_k \notin \Sigma_1 \wedge s_k \in \Sigma_2 \wedge q_j^2 \in Q_2 \\ (q_{\text{reject}}, q_{\text{reject}}) & \text{otherwise} \end{cases}$$

We will show that M recognizes L. First, we show that if $s \in L$, then M accepts s. Then, we show that if M accepts s, then $s \in L$.

Fix some $s \in L$ such that $s = s_1 s_2 \ldots s_m$. Define $r_0 = (q_0^1, q_0^2)$, and $r_{i+1} = \delta(s_i, r_i)$ for $0 \leq i \leq m-1$. Since $s \in L$, either $s \in L_1$ or $s \in L_2$. Assume $s \in L_1$. Then M_1 accepts s, so there exists some $r_0^1 = q_0^1$, $r_{i+1}^1 = \delta_1(s_i, r_i^1)$ for $0 \leq i \leq m-1$, where $r_m^1 \in F_1$. Consider the sequence of states $r_0, r_1, \ldots, r_m$

taken by M on s. Since the first component of r_0 is q_0^1, by definition of M, the first component of each r_i is r_i^1. Thus, the first component of r_m is r_m^1, so $r_m \in F_1 \times Q_2$. By definition, r_m is an accept state, so M accepts s. The case where $s \in L_2$ is similar.

Fix some s accepted by M such that $s = s_1 s_2 \ldots s_m$. Then, there exists some $r_0 = (q_0^1, q_0^2)$ and $r_{i+1} = \delta(s_i, r_i)$ for $0 \leq i \leq m-1$ such that $r_m = (q_j^1, q_k^2)$ where $q_j^1 \in Q_1$ or $q_k^2 \in Q_2$. Assume $q_j^1 \in Q_1$. By definition of M, there exists $r_0^1 = q_0^1$, $r_{i+1}^1 = \delta_1(s_i, r_i^1)$ for $0 \leq i \leq m-1$, such that $r_m^1 = q_j^1 \in Q_1$, namely r_i^1 is the first component of each r_i. Thus, M_1 accepts s, so $s \in L_1 \subseteq L$. The case where $q_k^2 \in Q_2$ is similar.

Thus M accepts s if and only if $s \in L$. Hence L is regular as desired. //

Part (b)

Prove or disprove: the class of regular languages is closed under concatenation.

Proof. Let L_1 and L_2 be regular languages. We will show that $L = L_1 \circ L_2$ is regular by construction. The main idea is to non-deterministically guess a split $s = ab$ such that $a \in L_1$ and $b \in L_2$.

Since L_1 is regular, there exists some NFA $M_1 = (Q_1, \Sigma_1, \delta_1, q_0^1, F_1)$ that recognizes L_1, and since L_2 is regular, there exists some NFA $M_2 = (Q_2, \Sigma_2, \delta_2, q_0^2, F_2)$ that recognizes L_2. Define NFA

$$M = (Q_1 \cup Q_2, \Sigma_1 \cup \Sigma_2, \delta, q_0^1, F_2)$$

where

$$\delta_1'(s_k, q_i) = \begin{cases} \delta_1(s_k, q_i) & \text{if } q_i \in Q_1 \wedge s_k \in \Sigma_1 \cup \{\epsilon\} \\ \emptyset & \text{otherwise} \end{cases}$$

$$\delta_2'(s_k, q_i) = \begin{cases} \delta_2(s_k, q_i) & \text{if } q_i \in Q_2 \wedge s_k \in \Sigma_2 \cup \{\epsilon\} \\ \emptyset & \text{otherwise} \end{cases}$$

$$\delta(s_k, q_i) = \begin{cases} \delta_1'(s_k, q_i) \cup \{q_0^2\} & \text{if } q_i \in F_1 \wedge s_k = \epsilon \\ \delta_1'(s_k, q_i) \cup \delta_2'(s_k, q_i) & \text{otherwise} \end{cases}$$

We will show that M recognizes L. First, we show that if $s \in L$, then M accepts s. Then, we show that if M accepts s, then $s \in L$.

Fix some $s \in L$ where $s = s_1 s_2 \ldots s_m$. Then, there exists $a \in L_1$ and $b \in L_2$ such that $s = ab$. Since $a \in L_1$, M_1 accepts a. Therefore, there exists some $r_0^1 = q_0^1$, $r_i^1 \in \delta_1(a_i, r_{i-1}^1)$ for $1 \leq i \leq |a|$, where $r_{|a|}^1 \in F_1$. Likewise, M_2 accepts b. Therefore, there exists some $r_0^2 = q_0^2$, $r_i^2 \in \delta_2(b_i, r_{i-1}^2)$ for $1 \leq i \leq |b|$, where

$r^2_{|b|} \in F_2$. Define $s' = a\epsilon b = s$, and $r = r_1 r_2 \dots r_{|s'|}$ where

$$r_i = \begin{cases} r_0^1 & \text{if } i = 0 \\ r_i^1 & \text{if } 1 \leq i \leq |a| \\ r_0^2 & \text{if } i = |a| + 1 \\ r_i^2 & \text{if } |a| + 2 \leq i \leq |s'| \end{cases}$$

Now, we must verify that r is an accepting computation of M on s':

1. We have that $r_0 = r_0^1 = q_0^1$ which is the start state of M.
2. We consider three subcases
   (a) Assume $1 \leq i \leq |a|$. Then
   $$r_i = r_i^1 \in \delta_1(a_i, r_{i-1}^1) = \delta_1'(s_i', r_{i-1}) \subseteq \delta(s_i', r_{i-1})$$
   (b) Assume $i = |a| + 1$. Then
   $$r_i = r_0^2 = q_0^2 \in \delta(\epsilon, r^1_{|a|}) = \delta(s'_{|a|}, r_{i-1})$$
   (c) Assume $|a| + 2 \leq i \leq |s'|$. Then
   $$r_i = r_i^2 \in \delta_2(b_i, r^2_{i-|a|-2}) = \delta_1'(s_i', r_{i-1}) \subseteq \delta(s_i', r_{i-1})$$
   so for all $1 \leq i \leq |s'|$, $r_i \in \delta(s_i', r_{i-1})$.
3. We have that $r_{|s'|} = r^2_{|b|} \in F_2$, so the final state is an accept state.

Therefore, M accepts s.

Fix some $s = s_1 s_2 \dots s_m$ where M accepts s.. Then, there exists some accepting sequence of states $r = r_1 r_2 \dots r_t \epsilon r_{t+2} \dots r_m$, where r must contain ϵ otherwise all $q \in Q_2$ are unreachable, which is impossible since $r_m \in F_2 \subseteq Q_2$. Define $a = s_1 \dots s_k$ and $b = s_{k+2} \dots s_m$. We must show that $a \in L_1$ and $b \in L_1$. First, we show that $a \in L_1$. Since

Thus M accepts s if and only if $s \in L$. Hence L is regular as desired. //

Part (c)

Prove or disprove: the class of regular languages is closed under star.

Proof. TODO //