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 $uvw = s$, $|uv| \leq p$ and $|v| \geq 1$, 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 $uvw = s$, $|uv| \leq p$ and $|v| \geq 1$, $v = 0^k$ for some $1 \leq k \leq p$ since the first p symbols in s are all zero. By the pumping lemma, $s' = uv^0w = 0^{p-k}1^p 0^p 1^p \in L$. Suppose there exists some choice of x', y' such that $s' = x'y'x'y'$. Then, the first half of s' is $x'y' = 0^{p-k}1^p 1^{k/2}$ 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 non-regular languages is closed under union.

Proof. We will show that the class of non-regular languages is not closed under union by counterexample. The main idea is that the complement of a non-regular language is non-regular, but the union of any language and its complement is regular.

First, we will show that class of non-regular languages is closed under complement. For the sake of contradiction, suppose not. Then, there exists some non-regular language L such that L^c is regular. However, since the class of regular languages is closed under complement, $(L^c)^c = L$ is regular, which is impossible. Therefore, the class of non-regular languages is closed under complement.

Now, we will show that the class of non-regular languages is not closed under union. Fix some non-regular language L. Then L^c is also non-regular. However, $L \cup L^c = \Sigma^*$ which is regular. //

Part (b)

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

Proof. We will show that the class of non-regular languages is not closed under concatenation by counterexample. The main idea is that the set of unary strings with perfect square lengths is non-regular, but any integer can be written as the sum of four perfect squares, so any unary string can be written as the concatenation of four strings of perfect square length.

Let $L = \{1^{n^2} \mid n \geq 0\}$ where $\Sigma = \{1\}$. We will show that L is non-regular via the pumping lemma. Assume that L is regular, and let p be the pumping length. Take $s = 1^{(p+1)^2}$.

We will show that s cannot be pumped. For any choice of u, v, w such that $uvw = s$, $|uv| \leq p$ and $|v| \geq 1$, $v = 1^k$ for some $1 \leq k \leq p$. By the pumping lemma, $s' = uv^0w = 1^{(p+1)^2-k} \in L$. Since $k \geq 1$, $(p+1)^2 - k \leq (p+1)^2 - 1 < (p+1)^2$. Since $k \leq p$, $(p+1)^2 - k \geq (p^2+2p+1) - p > p^2$. Since $p^2 < (p+1)^2 - k < (p+1)^2$, $(p+1)^2 - k$ is not a perfect square, so $s' \notin L$. Hence s cannot be pumped, so L is non-regular.

Now, we will show that the class of non-regular languages is not closed under concatenation. By Lagrange's four square theorem,

$$\begin{aligned} L \circ L \circ L \circ L &= \{1^{a^2} \mid a \geq 0\} \circ \{1^{b^2} \mid b \geq 0\} \circ \{1^{c^2} \mid c \geq 0\} \circ \{1^{d^2} \mid d \geq 0\} \\ &= \{1^{a^2}1^{b^2}1^{c^2}1^{d^2} \mid a, b, c, d \geq 0\} \\ &= \{1^{a^2+b^2+c^2+d^2} \mid a, b, c, d \geq 0\} \\ &= \{1^n \mid n \geq 0\} \\ &= \Sigma^* \end{aligned}$$

which is regular. //

Part (c)

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

Proof. We will show that the class of non-regular languages is not closed under star. The main idea is that non-regular languages remain non-regular under union with Σ, but then star generates Σ^*.

Let L be some non-regular language with alphabet Σ, and let $\Sigma^- = \Sigma - L$. Take $L' = L \cup \Sigma$. We will show that L' is non-regular by contradiction. For the sake of contradiction, suppose that L' is regular. Since Σ^- is finite, Σ^- is regular. Therefore, $(\Sigma^-)^c$ is regular. Since the class of regular languages is closed under intersection, $L' \cap (\Sigma^-)^c$ is regular. However,

$$\begin{aligned} L' \cap (\Sigma^-)^c &= (L \cup \Sigma^-) \cap (\Sigma^-)^c \\ &= (L \cap (\Sigma^-)^c) \cup (\Sigma^- \cap (\Sigma^-)^c) \\ &= (L \cap (\Sigma - L)^c) \cup \emptyset \\ &= L \end{aligned}$$

which is impossible. Therefore, L' is non-regular.

Now, we will show that the class of non-regular languages is not closed under star. We have that

$$(L')^* = (L \cup \Sigma)^* = (L^* \cup \Sigma^*)^* = (\Sigma^*)^* = \Sigma^*$$

which is regular. //

[Misread] 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 $s = s_1 s_2 \dots s_m$. Define $r_0 = (q_0^1, q_0^2)$, and $r_{i+1} = \delta(s_i, r_i)$ for $0 \le i \le m-1$. 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 $0 \le i \le m-1$, where $r_m^1 \in F_1$. Consider the sequence of states $r_0, r_1, \dots, r_m$ 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 $s = s_1 s_2 \dots s_m$. Then, there exists some $r_0 = (q_0^1, q_0^2)$ and $r_{i+1} = \delta(s_i, r_i)$ for $0 \le i \le m-1$ 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 $b \in L_2$. We start in the machine recognizing L_1, ϵ-move from the accept states in the machine recognizing L_1 to the start state in the machine recognizing L_2, and accept if the machine recognizing L_2 accepts.

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 $r_{|b|}^2 \in F_2$. Define $s' = a\epsilon b = s$, and $r = r_1 r_2 \ldots r_{|s'|}$ 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
   $$r_i = r_0^2 = q_0^2 \in \delta(\epsilon, r_{|a|}^1) = \delta(s_{|a|}', r_{i-1})$$
   since $r_{|a|}^1 \in F_1$.
   (c) Assume $|a| + 2 \leq i \leq |s'|$. Then
   $$r_i = r_i^2 \in \delta_2(b_i, r_{i-|a|-2}^2) = \delta_1'(s_i', r_{i-1}) \subseteq \delta(s_i', r_{i-1})$$
   so for all $1 \leq i \leq |s'|$, $r_i \in \delta(s_i', r_{i-1})$.
3. We have that $r_{|s'|} = r_{|b|}^2 \in F_2$, so the final state is an accept state.

Therefore, M accepts s.

Fix some $s = s_1 s_2 \ldots s_m$ where M accepts s. Then, there exists some accepting sequence of states $r = r_0 r_1 \ldots r_m$. Since $r_m \in F_2 \subseteq Q_2$, there exists $r_i \in Q_2$. Let r_{t+1} be the first such $r_i \in Q_2$. Then $r_t \in Q_1$ so $s_{t+1} = \epsilon$, since the states in Q_2 are only reachable from the states in Q_1 by ϵ-moves. Define $a = s_1 \ldots s_t$ and $b = s_{t+2} \ldots s_m$. Then $s = a\epsilon b = ab$. We must show that $a \in L_1$ and $b \in L_1$. First, we show that $a \in L_1$. Define

$$r^1 = r_0^1 r_1^1 \ldots r_t^1 = r_0 r_1 \ldots r_t$$

We must show that r^1 is an accepting sequence of states for M_1:

1. We have that $r_0^1 = r_0 = q_0^1$ which is the start state.
2. We have that $r_i^1 = r_i \in \delta(s_i, r_{i-1})$, and by choice of t, $r_{i-1} \in Q_1$, so
   $$\delta(s_i, r_{i-1}) = \delta_1'(s_i, r_{i-1}) = \delta_1(s_i, r_{i-1}) = \delta_1(s_i, r_{i-1}^1)$$
   Thus, $r_i^1 \in \delta_1(s_i, r_{i-1}^1)$.
3. We have that $r_t^1 = r_t \in F_1$. Because transitions from states in Q_1 to Q_2 can only occur from states in F_1, if $r_t \notin F_1$, $r_{t+1} \notin Q_2$ which is impossible. Thus, r_t is an accept state.

The proof that $b \in L_2$ proceeds similarly. Therefore, $s = a\epsilon b = ab \in L$.

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. Let L_1 be a regular language. We will show that $L = L_1^*$ by construction. The main idea is to non-deterministically split $s = a^1 a^2 \dots a^m$ where $a^1, a^2, \dots, a^m \in L_2$ (a^i is the i-th word, not a repeated i times). We define a new start state as the only accept state, and ϵ-move from the new start state to the old start state, and from each old accept state to the new start state.

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. Define NFA

$$M = (Q_1 \cup \{q_{\text{start}}\}, \Sigma_1, \delta, q_{\text{start}}, \{q_{\text{start}}\})$$

where

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

$$\delta(s_k, q_i) = \begin{cases} \{q_0^1\} & \text{if } q_i = q_{\text{start}} \wedge s_k = \epsilon \\ \delta_1'(s_k, q_i) \cup \{q_{\text{start}}\} & \text{if } q_i \in F_1 \wedge s_k = \epsilon \\ \delta_1'(s_k, q_i) & \text{otherwise} \end{cases}$$

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 $s \in L$. Since I'm out of time, and have (hopefully) sufficiently demonstrated that I can grind out DFA proofs, I will just give a proof sketch for both directions.

Fix some $s \in L$. Since $s \in L$, $s = a^1 a^2 \dots a^m$ where each $a^i \in L_1$. Since $a^i \in L_1$, there is an accepting sequence of states r^i in M_1. Define $r = r^1 r^2 \dots r^m$. Then, take $s' = \epsilon a^1 \epsilon \epsilon a^2 \epsilon \epsilon \dots \epsilon \epsilon a^m \epsilon = s$. It can be shown that r is an accepting sequence of states for M on s', so M accepts $s' = s$.

Fix some s where M accepts s. Since M accepts s, there is an accepting sequence of states $r = r_0 r_1 \dots r_m$. Split r into the maximum number of segments $r^1, r^2, \dots r^n$, such that $r = q_{\text{start}} r^1 q_{\text{start}} r^2 q_{\text{start}} \dots q_{\text{start}} r^n q_{\text{start}}$. Then, it can be shown that each $r^1, r^2 \dots r^n$ is an accepting sequence for M_1 given the corresponding substring in $a^1, a^2, \dots a^n$, where $s' = \epsilon a^1 \epsilon \epsilon a^2 \epsilon \epsilon \dots \epsilon \epsilon a^n \epsilon = s$. Thus $a^1, a^2, \dots, a^n \in L_1$, so $s \in L$.

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