CSC236 Homework Assignment #3

Language Regularity, Regular Expressions, and DFA/NFA Complexity

Alexander He Meng

Prepared for November 25, 2024

Question #1

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

(a):

<u>Claim:</u> Σ^* is a regular language.

Proof.
Let $L_1 = \{0\}$ and $L_2 = \{1\}$ be regular languages of Σ.
Define $L_3 = L_1 \cup L_2 = \{0, 1\}$ as the regular language obtained by the union of L_1 and L_2.

By definition, ${L_3}^*$ is a regular language. Since $L_3 = \Sigma$, Σ^* is also a regular language.

Next, denote the transition function δ by the following table:

Old State	**Symbol**	**New State**
q_0	0	q_0
q_0	1	q_0

Table 1: State Transition Table

Using δ, define the deterministic finite automaton $\mathcal{D} = (\mathcal{Q}, \Sigma, \delta, s, F)$, where

$$\mathcal{Q} = \{q_0\} \text{ is the set of states in } \mathcal{D}$$

$$\Sigma = \{0, 1\} \text{ is the alphabet of symbols used by } \mathcal{D}$$

$$\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q} \text{ is the transition function defined by Table 1}$$

$$s = q_0 \text{ is the initial state of } \mathcal{D}$$

$$F = \{q_0\} \subseteq \mathcal{Q} \text{ is the set of accepting states of } \mathcal{D}.$$

$\square$

(b):

<u>Claim:</u> $\Sigma^* \setminus K, K = \{01, 101, 010\}$ is a regular language.

Proof.

This proof aims to show that the language $\Sigma^* \setminus K$ is a regular language by constructing a DFA that accepts all strings except the literal strings in $K = \{01, 101, 010\}$.

Denote the transition function δ by the following table: Using δ, define the DFA $\mathcal{D} =$

Old State	**Symbol**	**New State**
q_ϵ	0	q_0
q_ϵ	1	q_1
q_0	0	q_5
q_0	1	q_2
q_1	0	q_4
q_1	1	q_5
q_2	0	q_3
q_2	1	q_5
q_3	0	q_5
q_3	1	q_5
q_4	0	q_5
q_4	1	q_3
q_5	0	q_5
q_5	1	q_5

Table 2: State Transition Table

$(\mathcal{Q}, \Sigma, \delta, s, F)$, where

$$\mathcal{Q} = \{q_0\} \text{ is the set of states in } \mathcal{D}$$

$$\Sigma = \{0, 1\} \text{ is the alphabet of symbols used by } \mathcal{D}$$

$$\delta : \mathcal{Q} \times \Sigma \to \mathcal{Q} \text{ is the transition function define by Table 2}$$

$$s = q_0 \text{ is the initial state of } \mathcal{D}$$

$$F = \{q_2, q_3\} \subseteq \mathcal{Q} \text{ is the set of accepting states of } \mathcal{D}.$$

For each state in $\mathcal{D}$, define a state invariant $P_q(w)$, describing the property of any string $w \in \Sigma^*$ which leads $\mathcal{D}$ to state q:

$$P_{q_\epsilon}(w) :$$

$$P_{q_0}(w) :$$

$$P_{q_1}(w):$$

$$P_{q_2}(w):$$

$$P_{q_3}(w):$$

$$P_{q_4}(w):$$

$$P_{q_5}(w):$$

□

(c):
Claim: $\{w|w \text{ is a palindrome}\}$ is NOT a regular language.

Proof.
proofgoeshere

□

(d):
Claim: $\{ww|w \in \Sigma*\}$ is NOT a regular language.

Proof.
proofgoeshere

□

(e):
Claim: $\{w|ww \in \Sigma*\}$ is a regular language.

Proof.
proofgoeshere

□

(f):
Claim: $\{w|w \text{ is a binary representation of a multiple of } 3\}$ is a regular language.

Proof.
proofgoeshere

□

Question #2

<u>**Claim:**</u> Regular expressions that also have access to complement can still only express the same class of languages (i.e. the class of regular langauges) as regular expressions without the complement operation.

Proof.
Suppose r is an arbitrary regular expression with alphabet Σ so that $L = \mathcal{L}(r)$ is a regular language. By definition, there exists a DFA $\mathcal{M}$ that accepts L.

Construct $\mathcal{M}$:
$\mathcal{M} = (Q, \Sigma, \delta, s, F)$, where

- Q is a set of finite states;
- Σ is the alphabet;
- $\delta : Q \times \Sigma \to Q$ is the transition function;
- s is the start state;
- $F \subseteq Q$ is the set of accepting states.

Next, define $\overline{L} = \Sigma^* \setminus L$ as the regular language containing everything obtainable from the alphabet Σ except members of L.

Construct $\overline{M}$ to be identical to M, except for its accepting states:

- Q is a set of finite states;
- Σ is the alphabet;
- $\delta : Q \times \Sigma \to Q$ is the transition function;
- s is the start state;
- $Q \setminus F \subseteq Q$ is the set of accepting states.

Notice that $\overline{M}$ share the same states, alphabet, transition function, and start state as those of M. The difference is the set of accepting states in $\overline{M}$, which is designed to be mutually exclusive from that of M.

Self-Note: now add the proof that $\overline{M}$ accepts $\overline{L}$, and connect it back to regexes.
Next, let $\bar{r}$ be some regular expression without the complement operation.

$\square$

Question #3

Counter-free languages are a subset of languages that satisfy the condition:

$$(\exists n \in \mathbb{N})(\forall m \geq n)(xy^m z \in L \iff xy^n z \in L).$$

Star-free regular expressions are regular expressions without the Kleene star, but with complementation.

It is known in formal language theory that counter-free languages are equivalent to the languages that can be expressed as **star-free regular expressions**.

(a):

<u>**Claim:**</u> $(ab)*$ can be matched with a star-free regular expression, where $\Sigma = \{a, b\}$.

Proof.
The expression $(ab)*$ represents strings in the set $\{\epsilon, ab, abab, ababab, \dots\}$.

This means matching strings are strings where every occurrence of a is immediately followed by a b.

Due to the definition of $\Sigma = \{a, b\}$, an equivalent star-free regular expression can be written in terms of the complement. The complement definition is highlighted below:

- The string starts with b if it is not empty;
- The string contains an a not immediately followed by a b.

$\square$

(b):

<u>**Claim:**</u> $(ab)*$ is not a counter-free language, where $\Sigma = \{a, b\}$.

Proof.
proofgoeshere

$\square$

(c):

<u>Claim:</u> $(aa)*$ is not a counter-free language, where $\Sigma = \{a\}$.

Proof.
proofgoeshere

□

Question #4

Consider the language $L = \{w | \text{the third last character of } w \text{ is } 1\}$.

Let $k \in \mathbb{N}$ be arbitrary.

(a):

Claim: A DFA that accepts L has to have at least 2^k number of states.

Proof.
proofgoeshere

□

(b):

Claim: The smallest NFA that accepts L has to have exactly k number of states.

Proof.
proofgoeshere

□

(c):

Claim: The smallest DFA that accepts L has to have exactly $2^{k+1} - 1$ number of states.

Proof.
proofgoeshere

□

Question #5

<u>Claim:</u> Every finite language can be represented by a regular expression (meaning all finite languages are regular).

Proof.
proofgoeshere

□