Problem 1. For each of the following languages (with alphabet \{0, 1\}), give a regular expression for it. For the more complicated languages, it may help to collapse a GNFA for the language to a single transition.

1. Any string ending in 110.

2. Any string which can be broken into a (non-overlapping) sequence of the following strings: 000, 111, 101, or 010.

3. Any string of even length starting with 0 or any string of odd length starting with 1.

4. Any string not containing the substrings 11 or 000.

5. Any string except 101010.

Solution 1.

a) \((0|1)^*110\)

b) \((000|111|101|010)^*\)

c) \(0(0|1)(0|1)(0|1)^*1(0|1)(0|1)^*\)

d) \(\epsilon[0|00](1|01|001)(01|001)^*(\epsilon|0|00)\)

e) \(\epsilon[0|1|00|01|10|11|000|\ldots|101001|101011|101100|\ldots|111111](0|1)^7(0|1)^*\)

Problem 2. Let \(A\) be language of all even length strings of \(\{0, 1\}\). Note that \(A\) is regular.

a) If \(A \circ A^R\) is regular, prove it.

b) If \(\hat{A} = \{\omega\omega^R \mid \omega \in A\}\) is regular, prove it.

c) Give a brief intuitive explanation in your own words of why one is regular while the other is not.

Solution 2.

a) Let \(B\) be any regular language. Since the class of regular languages is closed under concatenation and reversal, \(B \circ B^R\) is regular. Since \(A\) is regular, we’re done. \(\square\)

b) This language is not regular.
c) $A \circ A^R$ only requires that we have a string from $A$ and then a reversed string from $A$. Recognizing either is not hard. On the other hand, $\hat{A}$ requires that we remember a string of unbounded length so we can recognize its reversed form. Finite automata obviously can’t hold more than a finite amount of memory, so this is impossible.

**Problem 3.** We characterized a regular expression $R$ over an alphabet $\Sigma$ as matching one of the following (sometimes recursive) conditions:

1. $R = a$ for some $a \in \Sigma$
2. $R = \epsilon$
3. $R = \emptyset$
4. $R = R_1 | R_2$, where $R_1$ and $R_2$ are both regular expressions
5. $R = R_1 \circ R_2$, where $R_1$ and $R_2$ are both regular expressions
6. $R = R_1^*$, where $R_1$ is a regular expression.

Other expressions can be characterized in a similar manner. One major example is the arithmetic expressions. For example, if $E$ is an arithmetic expression, we would expect $pEq$ to also be an arithmetic expression but not $pE$ or $Eq$.

Define the alphabet $\Sigma = \mathbb{Z}_{10} \cup \{+,-,*,,/.,(,\})$ ($-$ here may be binary or unary).

a) Give a characterization similar to the characterization of regular expressions above for the arithmetic expressions.

b) Explain why the arithmetic expressions are not regular.

**Solution 3.**

a) We characterize the arithmetic expressions as follows. We say that $A$ is an arithmetic expression if $A$ is

1) some $\omega \in 0|\mathbb{Z}_{10}\backslash\{0\}|\mathbb{Z}^*_0$,
2) $A_1 + A_2$ where $A_1$ and $A_2$ are arithmetic expressions,
3) $A_1 - A_2$ where $A_1$ and $A_2$ are arithmetic expressions,
4) $A_1 \times A_2$ where $A_1$ and $A_2$ are arithmetic expressions,
5) $A_1/A_2$ where $A_1$ and $A_2$ are arithmetic expressions,
6) $(A_1)$ where $A_1$ is an arithmetic expression, or
7) $-A_1$ where $A_1$ is an arithmetic expression.

b) Parenthesis matching requires remembering the number of parentheses. This number is unbounded, so a finite automaton cannot match them properly.

**Problem 4.** Given the DFA below, prove that the states $q_0$ and $q_1$ are distinguishable.
Solution 4. Let the DFA be \((Q, \Sigma, \delta, q_0, F)\). It suffices to find a string \(\omega \in \Sigma^*\) such that \(\delta^*(q_0, \omega) \in F\) and \(\delta^*(q_1, \omega) \notin F\). Pick \(\omega = 1\). Then \(\delta^*(q_0, \omega) = q_1 \in F\) but \(\delta^*(q_1, \omega) = q_0 \notin F\).

Problem 5. Recall that an equivalence relation is a binary relation between a set and itself that is reflexive, symmetric, and transitive. An equivalence class is a set of objects in a relation that are all equal to each other. Lastly, a family of subsets of a set \(S\) partitions \(S\) if their union is \(S\) and they are pairwise disjoint.

Suppose the DFA \(D\) below is given by the 5-tuple \((Q = \{1\}, \delta, q_0, F)\).

![DFA Diagram]

Prove the following.

1. The sets \(Q_1 = \{q_0, q_2\}\) and \(Q_2 = \{q_1, q_3\}\) partition \(Q\).
2. Indistinguishably, written \(q =_D p\) if states \(q, p \in Q\) are indistinguishable, is an equivalence relation.
3. The sets \(Q_1\) and \(Q_2\) are the equivalence classes of \(=_D\) (you need only give a persuasive argument here, as a full proof is tedious beyond belief).

Solution 5. First, clearly \(Q_1\) and \(Q_2\) partition \(Q\) by inspection.

Next, consider \(q \in Q\). Clearly, for any string \(\omega \in \Sigma^*\), \(\delta^*(q, \omega) \in F\) if and only if \(\delta^*(q, \omega) \in F\). Thus \(q =_D q\) and therefore \(=_D\) is reflexive.

Now consider \(q, p \in Q\) such that \(q =_D p\). For any string \(\omega \in \Sigma^*\), we have \(\delta^*(q, \omega) \in F\) if and only if \(\delta^*(p, \omega) \in F\). This is equivalent to the statement \(\delta^*(p, \omega) \in F\) if and only if \(\delta^*(q, \omega) \in F\). Thus \(p =_D q\), and therefore \(=_D\) is symmetric.

Next consider \(r, s, t \in Q\) such that \(r =_D s\) and \(s =_D t\). Then for any string \(\omega \in \Sigma^*\), we have \(\delta^*(r, \omega) \in F\) if and only if \(\delta^*(s, \omega) \in F\) and \(\delta^*(s, \omega) \in F\) if and only if \(\delta^*(t, \omega) \in F\). We can combine these statements by cutting out the middle state, yielding \(\delta^*(r, \omega) \in F\) if and only if \(\delta^*(t, \omega) \in F\). Thus \(r =_D t\) and therefore \(=_D\) is transitive.

As a consequence of these three properties, \(=_D\) is an equivalence relation.
Lastly, it remains to show that \( Q_1 \) and \( Q_2 \) are the equivalence classes of \( =_D \). To do so, we must show two things. First, within any set, the elements are indistinguishable. Second, two states from different sets are distinguishable.

For a string \( \omega \in \Sigma^* \), we have that \( \omega = 1^n \) for some nonnegative integer \( n \). By inspection, we can see that for any \( q_j \),

\[
\delta^*(q_j, 1^n) = q_i,
\]

where \( i \equiv n + j \mod 4 \). So clearly \( \delta^*(q_0, 1^n) \in F \) iff \( \delta^*(q_2, 1^n) \in F \), which makes \( q_0 =_D q_2 \). Similarly, we have \( \delta^*(q_1, 1^n) \in F \) iff \( \delta^*(q_3, 1^n) \in F \), which makes \( q_1 =_D q_3 \). This proves the first claim.

Lastly, if we consider two states \( q_i \) and \( q_j \) not in the same set, we obtain \( \delta^*(q_i, 1^n) \in F \) iff \( \delta^*(q_j, 1^n) \notin F \), which makes \( q_i =_D q_j \). This can be seen by inspection since \( i \) and \( j \) are not congruent modulo 2, and the DFA simply alternates between accepting and rejection. Thus the sets \( Q_1 \) and \( Q_2 \) also satisfy the second claim, and we are done. \( \square \)