DUE: Friday, September 13

1. (12 points) For any nonempty string \(a_1a_2...a_n\), we define the operation \(\text{shift}\) of \(a_1a_2...a_n\) as \(\text{shift}(a_1a_2...a_n) = a_2...a_na_1\), and we define the \(\text{shift}\) of a language \(L\) as \(\text{shift}(L) = \{\text{shift}(z) \mid z \in L\}\). So \(\text{shift}(\emptyset) = \emptyset\) and \(\text{shift}(\{a,b,abba\}) = \{a,b,bbaa\}\). If \(L\) is regular and \(\varepsilon \notin L\), must \(\text{shift}(L)\) be regular? Justify your answer. If your answer is that \(\text{shift}(L)\) is regular, then you only need show how to construct an automaton to accept it; you don’t need to prove that the automaton accepts \(\text{shift}(L)\).

2. (2 points) Give a regular expression for the set of positive integers, without leading 0’s, in decimal notation which are divisible by 5. So for example, 5 and 5705 and 10 belong to our language, but 17 and 0565 and 0 do not.

3. (12 points) State whether or not each of the following three conjectures are true for any NFA \(N = (Q, \Sigma, \Delta, S, F)\), and justify each response.

**Conjecture A:** If \(L(N) \neq \emptyset\), then \((\exists z \in L(N))(|z| < |Q|)\).

**Conjecture B:** If \(L(N) \neq \Sigma^*\), then \((\exists z \in L(N))(|z| < 2^{\|Q\|})\).

4. (8 points) We define an operator on languages by \(\text{Prefix}(L) = \{y \mid \exists z \ yz \in L\}\). So, for example, \(\text{Prefix}(\{011, 1\}) = \{\varepsilon, 0, 01, 011, 1\}\). Describe an algorithm to accept as input a regular expression \(\alpha\) and which constructs a regular expression for \(\text{Prefix}(L(\alpha))\). So upon receiving input 011+1 your algorithm would return \(\varepsilon + 0 + 1 + 01 + 011 + 1\), or an equivalent regular expression.

5. (5 points) Which of the following "identities" are true for all regular expressions \(\alpha, \beta\) and \(\gamma\)? If an "identity" is false, provide a counterexample.

- \(a\) \(\alpha(\beta \cup \gamma) = \alpha\beta \cup \alpha\gamma\)
- \(b\) \((\alpha^*)^* = \alpha^*\)
- \(c\) \((\alpha^* \beta^*)^* = (\alpha^* \cup \beta^*)^*\)
- \(d\) \(\alpha\alpha = \alpha\)
- \(e\) \((\alpha \cup \beta)^* = \alpha^* \cup \beta^*\)
1. Given a DFA to accept $L$, we convert it to a new NFA to accept $\text{shift}(L)$ by the following intuitive steps:

- store the first symbol read in a "register" which can contain any symbol of $\Sigma$, and then go to a state $q_{\text{first}}$ that allows the automaton to consume the first symbol while storing it in the "register",

- consume the input string while behaving like $M$,

- if the DFA is in a final state, and the next symbol read is the same as the symbol in the "register", then guess that the next symbol being read is the last symbol of the input string and go to the final state of the NFA.

So given a DFA $(Q,\Sigma,\delta,s,F)$ to accept $L$, we construct the NFA without $\varepsilon$-transitions which contains $Q$, new start state $s^*$, new final state $f^*$, and a new state $q_{\text{first}}$.

$$
\{(\Sigma \times Q) \cup \{s^*,f^*,q_{\text{first}}\},\Sigma,\Delta,\{s^*\},\{f^*\}\cup\{(a,q_{\text{first}})\mid \delta(s,a) \in F\}\}
$$

where being in state $(a,q), a \in \Sigma, q \in Q$, "means" that the first symbol read was an $a$ and the DFA would be in state $q$ having consumed the input already read.

$$
\Delta(s^*,a) = \{(a,q_{\text{first}})\}, \forall a \in \Sigma
$$

$$
\Delta([a,q_{\text{first}}],b) = \{(a,\delta(\delta(s,a),b))\}, \forall a,b \in \Sigma
$$

$$
\Delta([a,q],b) = \{(a,\delta(q,b))\}, \forall a,b \in \Sigma, \forall q \notin F
$$

$$
\Delta([a,q],b) = \{(a,\delta(q,a)),f^*\}, \forall a \in \Sigma, \forall q \in F
$$

$$
\Delta(f^*,a) = \emptyset, \forall a \in \Sigma
$$

2. $(1+2+3+4+5+6+7+8+9)(0+1+2+3+4+5+6+7+8+9)^* (0+5) + 5$

3. **Conjecture A** is true. If $L(N) \neq \emptyset$, then there is a path from a state of $S$ to a state of $F$. A shortest such path passes through at most $|Q|$ states (we can remove all cycles from any such path until every state is visited at most once), and this shortest path traverses fewer than $|Q|$
edges. The labels of the edges of this path comprise a word of length less than \(|Q|\) which belongs to \(L(N)\).

**Conjecture B** is true. Using the algorithm from class or from our text, we know there exists a DFA \(M=(Q', \Sigma, \delta, s, F')\) such that \(L(M)=L(N)\) and \(|Q'|=2^{k_1}\). If \(L(N) \neq \Sigma^*\), then for some \(z \in \Sigma^*\) the path labeled \(z\) leads to a state not in \(F\). Let \(z\) be a shortest such string. If \(|z|<|Q'|=2^{k_1}\), then we are done. Otherwise the path corresponding to \(z\) passes through some state at least two times. Removing the substring of \(z\) corresponding to the cycle, we obtain a shorter string not leading to a state of \(F\), contradicting the fact that \(z\) was a shortest string.

4. Recursively apply the following function \(\text{Prefix}\) recursively to regular expression \(\alpha\).

\[
\text{Prefix}(\alpha) =
\begin{cases} 
\emptyset & \text{if } \alpha = \emptyset \\
\epsilon & \text{if } a \in \Sigma \\
\text{Prefix}(\beta) + \text{Prefix}(\gamma) & \text{if } \alpha = \beta + \gamma \\
\text{Prefix}(\beta) + \beta \text{Prefix}(\gamma) & \text{if } \alpha = \beta \gamma \\
\beta^* \text{Prefix}(\beta) & \text{if } \alpha = \beta^*
\end{cases}
\]

5. \(a\) true
\(b\) true
\(c\) true
\(d\) false Let \(\alpha = 0\). \(\alpha \alpha = 0 0 \neq 0 = \alpha\)
\(e\) false Let \(\alpha = 0\) and \(\beta = 1\). \(0 \in (\alpha \cup \beta)^* = (\alpha \cup \beta)^*\) though \(0 \in \epsilon \alpha^* \cup \beta^* = 0^* \cup 1^*\)