There are 6 numbered questions. The 6 parts of Question 1 are worth 4 points each. Questions 2 through 6 are worth 15 points each. You get one point for free.

1. Which languages are regular? You don’t need to prove your answers. Write an “R” in the blank next to the description of each language you think is regular. Write “N” for any language you think is not regular. In each case the alphabet is $\Sigma = \{0,1\}$

   a. __R__ Strings that end in exactly 10 1’s. So 010111111111 is in this language but 011111111111 is not.

   b. __R__ Strings with any number of 0’s followed by an even number of 1’s.

   c. __R__ Strings where the digits sum to a number divisible by 5.

   d. __N__ Strings where there are at least as many 0’s as 1’s.

   e. __R__ $0^* \mathcal{L}$ (that is the concatenation of two languages), where $\mathcal{L} = \{0^n \mid n \text{ is prime}\}$   $0^* \mathcal{L} = \{0^n \mid n > 1\}$

   f. __R__ Strings of length 1000 that have a prime number of 1’s. This is a finite language.
2. Here is an ε-NFA, with start state A.
   a) Convert this NFA to a DFA
   b) Describe in English the strings it accepts.

   Answer:

   This accepts all strings ending in 0.
3. Suppose we know that for some language $L$ we know that the language $00L = \{00\alpha \mid \alpha \in L\}$ is regular. Must $L$ be regular? Either give an example where $L$ is not regular and $00L$ is regular, or else show that $L$ must be regular if $00L$ is.

The language $L$ must be regular. Suppose $P = (\Sigma,Q,\delta,s,F)$ is a DFA accepting $00L$. Let $q = \delta(s,0)$ and let $q_1 = \delta(q,0)$. State $q_1$ is where you get to in $P$ on input $00$. Let $P' = (\Sigma,Q,\delta,q_1,F)$. $P'$ is the same as $P$ only with start state $q_1$. Now suppose string $\alpha$ is in $L$. Then $00\alpha$ is in $00L$ and takes $P$ from state $s$ to $q$ to $q_1$ and then eventually to a final state. So $\alpha$ takes $P'$ from $q_1$ to a final state, and $P'$ accepts $\alpha$. Similarly, if $\alpha$ takes $P'$ from $q_1$ to a final state then $00\alpha$ takes $P$ from $s$ to a final state, so $00\alpha$ is in $00L$ and $\alpha$ must be in $L$. Altogether, the DFA $P'$ accepts $\alpha$ if and only if $\alpha$ is in $L$, so $L$ is regular.
4. Consider the following DFA. We had an algorithm for converting a DFA to a regular expression. This involved making a table of regular expressions $r_{ij}^k$.

Here is the first column of a table of the $r_{ij}^k$ expressions; find the 4 entries of the second column.

\[
\begin{array}{|c|c|c|}
\hline
i & k=0 & k=1 \\
\hline
r_{11}^k & \varepsilon+1 & 1^* \\
\hline
r_{12}^k & 0 & 0+(\varepsilon+1)1^*0=1^*0 \\
\hline
r_{21}^k & 1 & 1+11^*(\varepsilon+1)=1^* \\
\hline
r_{22}^k & \varepsilon+0 & \varepsilon+0+11^*0=\varepsilon+1^*0 \\
\hline
\end{array}
\]

\[
r_{ij}^1 = r_{ij}^0+r_{i1}^0(r_{11}^0)^*r_{1j}^0
\]
5. Use the pumping lemma to show carefully that the language \( \{0^m1^n0^n \mid m \geq 2, \ n \geq 0\} \) is not regular.

Suppose this language is regular. Let \( p \) be its pumping constant. Let \( w = 0^21^p0^p \). This is a string longer than \( p \), so it should be pumpable. Suppose \( w = xyz \) is any decomposition of \( w \) with 
\[
|xy| \leq p \text{ and } y \text{ nonempty.}
\]
If \( y \) contains any 0s then \( xy^0z \) has fewer than 2 leading 0s and so is not in the language. If \( y \) contains any 1s then \( xy^2z \) has more 1s than trailing 0s and so is not in the language. This means there is no pumpable decomposition of \( w \), and the language can't be regular.
6. Give a grammar for the language \( \{ 0^n1^m \mid n > m > 0 \} \)

I think of this language as \( 0^+ \{ 0^m1^m \mid m > 0 \} \). Here is a grammar for that way of thinking of it:

\[
S \rightarrow AB \\
A \rightarrow 0A | 0 \quad (A \text{ generates } 0^+) \\
B \rightarrow 0B1 | 01 \quad (B \text{ generates } \{ 0^m1^m \mid m > 0 \})
\]