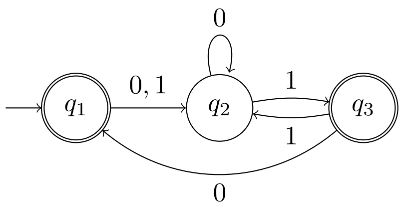
CMSC 303 Introduction to Theory of Computation, VCU
Assignment: 3
Name: Steven Hernandez

1. (a) $R_a = 0\Sigma^*1$
   Which says: 0 concatenated with zero or more character concatenated with 1.
   (b) $R_b = (\Sigma^*0\Sigma^*)^4$
   Says: zero or more characters followed by a 0 follower by zero or more of any character, which is then repeated 4 times.
   (c) $R_c = 1 \bigcup 11 \bigcup \epsilon$
   Which explicitly states the contents of the language.
   (d) $R_d = \{\Sigma\} \bigcup \{\Sigma\Sigma\} \bigcup \{\Sigma\Sigma\Sigma\} \bigcup \{\epsilon\}$
   Explicitly allows for any strings with one character or two characters or three characters of no characters.
   (e) We develop this by beginning with a DFA M such that $M = \{x|x \text{ contains } 110\}$. From here, we change $F = Q - F$. Now, our DFA is matching $M = \{x|x \text{ contains } 110\}$.
   $R_e = \epsilon \bigcup 0^*(10)^*1 \bigcup 0^*(10)^*111^* = 0^*(10)^*1^*$
   The center group $(10)^*$ states that there must be a 0 following directly after any 1, or else we end up int the third part 1^*, which prevents us from placing another zero, or else we our language would allow $\Sigma^*110\Sigma^*$.
   (f) $R_f = \Sigma^+$
   Plus indicates 1 or more.

2. (a) $M_a = (Q, \Sigma, \delta, q, F)$ such that:
   $Q = \{q_0\}$
   Σ is our language
   $q = q_0$
   $F = \{q_0\}$
   $\delta = \epsilon$
   because any transitions would mean a character was read, which would not be a part of the language we are looking for.
   (b) $M_b = (Q, \Sigma, \delta, q, F)$ such that:
   $Q = \{q_0, q_1, q_2, q_3\}$
   $q = q_0$
   $F = \{q_3\}$
   Define δ by:

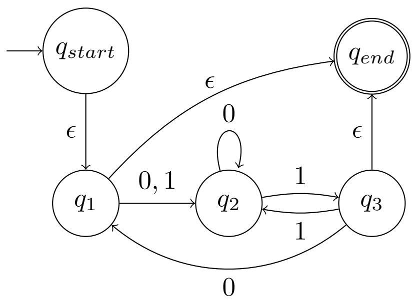
δ	0	1
q_0	q_0	q_1
q_1	q_0	q_2
q_2	q_0	q_3
q_3	q_3	q_3

3. State Diagram for M:

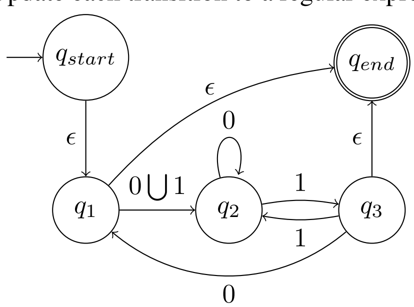


Steps for reaching regular expression for M:

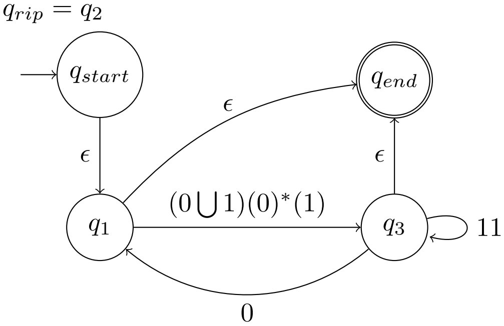
(a) Add q_{start} and q_{end} as explained in Lemma 1.60



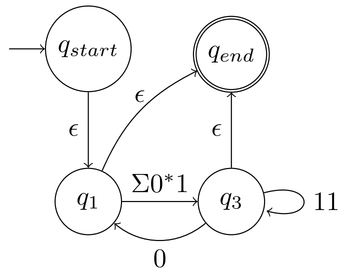
(b) Update each transition to a regular expression.



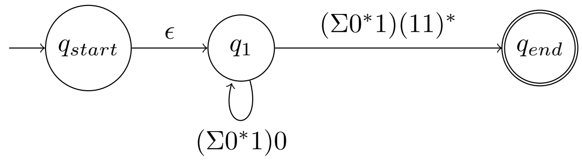
(c) $q_{rip} = q_2$



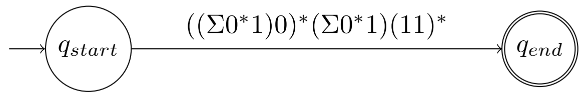
(d) Simplified to:



(e) $q_{rip} = q_3$



(f) $q_{rip} = q_1$



Thus our regular expression is $((\Sigma 0^*1)0)^*(\Sigma 0^*1)(11)^*$.

4. (a) Claim: $L = \left\{www | w \in \{0,1\}^*\right\}$ is not regular.

   Proof: Assume to the contrary that L is regular. Let p be the pumping length given by the pumping lemma. We choose $s = yyy$ such that $|y| = p$ and $y \in \{0,1\}^*$.

   By condition 3 of the pumping lemma, $|xy| \leq p$, we can conclude that $|xy| \leq |y|$. Thus, by pumping down s by setting $i = 0$, no matter the values for x, y, z we will be creating a new string $s_2 = y`yy$ such that $|y`| < |y|$. Thus $s_2 \notin L$. Contradiction.

   (b) Claim: $L = \{1^n 0^m 1^n | m, n \geq 0\}$ is not regular.

   Proof: Assume to the contrary that L is regular. Let p be the pumping length given by the pumping lemma. We choose $s = 1^p 0^1 1^p$. Notice, because $s \in L$ and $|s| > p$, the pumping lemma guarantees that s can be split into three peices $s = xyz$, where for any $i \geq 0$ the string $xy^i z$ is in L.

   Because by the condition 3 of the pumping lemma $|xy| \leq p$, both x and y can only contains 1s from the beginning of the string s. Thus, $x = 1^*$, $y = 1^+$ (because of condition 2, $|y| > 0$) and $z = 1^*01^p$. So, if we were to change i from $s = xy^i z$ such that $i = 0$, then out new string would become $1^{p-|y|}0^1 1^p \notin L$. Contradiction.

   (c) Claim: $L = \left\{x | x \in \{0,1\}^* \text{is not a palindrome}\right\}$ is not regular.

   Proof: We begin by creating a new language $L_2 = \left\{s | s \in \{0,1\}^* \text{is a palindrome}\right\}$. Note, this language is the inverse of L. Understand that if we have some DFA M, with F equal to the final states of M, then we can get $M`$ (inverse of M) by taking $Q - F$ from the original M. Thus, if we are able to show that L_2 is not regular, we can show as well that L is not regular because DFAs for each language would be the exact same other than the values for F for each language.

   Now, assume to the contrary, that L_2 is regular. Let p be the pumping length given by the pumping lemma. We choose a string $s = 1^p 0^1 1^p$. Notice, because $s \in L$ and $|s| > p$, the pumping lemma guarantees that s can be split into three peices $s = xyz$, where for any $i \geq 0$ the string $xy^i z$ is in L.

   Because by the condition 3 of the pumping lemma $|xy| \leq p$, both x and y can only contains 1s from the beginning of the string s. Thus, $x = 1^*$, $y = 1^+$ (because of condition 2, $|y| > 0$) and $z = 1^*01^p$. So, if we were to change i from $s = xy^i z$ such that $i = 0$, then out new string would become $1^{p-|y|}0^1 1^p \notin L_2$. Contradiction.

   As stated above, because L_2 is not regular and is the inverse of L, L is not regular either.

5. (a) The problem here, is that the pumping lemma does not allow us to select the values for x, y, z explicitely. We take into account all different possibilities for these variables.

   (b) Claim: $B = \left\{0^k x 0^k | k \geq 1 \text{ and } x \in \Sigma^*\right\}$ is regular.

   Notice: we don't really have to be aware of anything other than beginning and ending with a zero because $x \in \Sigma^*$ handles any discrepincies in number of zeros. So for example, $0010 \in B$ where $k = 1$ and $x = 01 \in \Sigma^*$.

   Proof: We construct an NFA $M = (Q, \Sigma, \delta, q, F)$ that recognizes B such that:

   $Q = \{q_0, q_1, q_2\}$
   $q = q_0$
   $F = \{q_2\}$
   Define δ by:

δ	0	1
q_0	$\{q_1\}$	$\emptyset$
q_1	$\{q_1, q_2\}$	$\{q_1\}$
q_2	$\emptyset$	$\emptyset$