Assignment 1

Due Friday Feb 3rd, 2017

Submission Instructions: Submit solutions in a single PDF via OWL. Assignments are due at 11:59:59 pm (Eastern Time) on the date listed above. Assignments submitted more than two days late will not be accepted and a mark of zero (0) will be recorded. See the course outline for details.

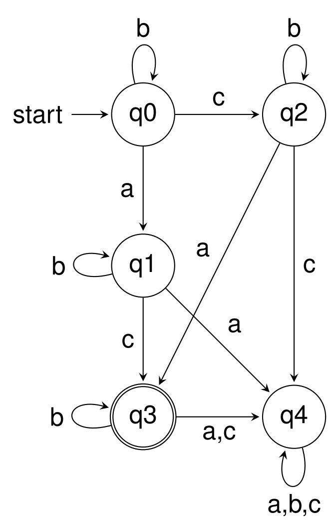
1. [6 marks] Describe the language

The formal description of a DFA M is $(S = \{q_0, q_1, q_2, q_3, q_4\}, \Sigma = \{a, b, c\}, \delta, start = q_0, F = \{q_3\})$, where δ is given by the following table:

	a	b	c
q_0	q_1	q_0	q_2
q_1	q_4	q_1	q_3
q_2	q_3	q_2	q_4
q_3	q_4	q_3	q_4
q_4	q_4	q_4	q_4

(a) [4 marks] Give the state diagram of this machine.

Answer:



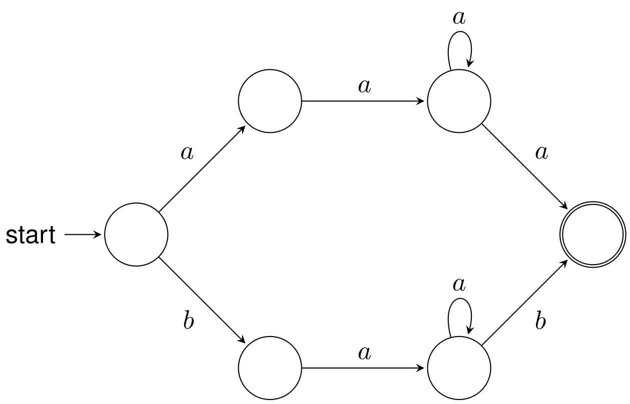
(b) [2 marks] Using *set builder* notation, describe the language accepted by M.

Answer: Accepts string that has exactly one 'A' and one 'C'

2. [8 marks] Identify the regular languages

For each of the following languages state whether it is regular or not. If L_i is regular, prove it by drawing a DFA or NFA (your choice) that recognizes it. If the language is not regular, give an argument (in plain English) why there is no DFA or NFA that can recognize it:

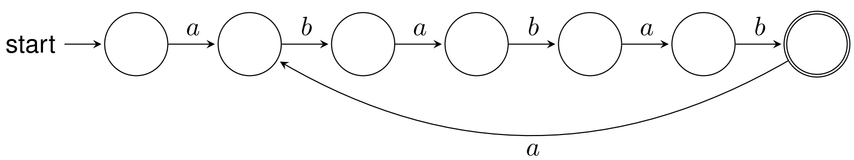
(a) [2 marks] $L_a = \{w(a^n)w : w \in \{a, b\}, n > 0\}$
Answer:



(b) [2 marks] $L_b = \{wabw : w \in \{a, b\}^*$ and $|w| > 2\}$
Answer: Not regular. The finite automaton would have to remember w to check that it appears twice, but w is unbounded suggesting a non-finite number of states to account for all possible w.

(c) [2 marks] $L_c = \{wwx : w, x \in \{a, b\}^*\}$ and $|x| > 3\}$
Answer: Not regular. Same reason as above.

(d) [2 marks] $L_d = \{(ababab)^n : n > 0\}$



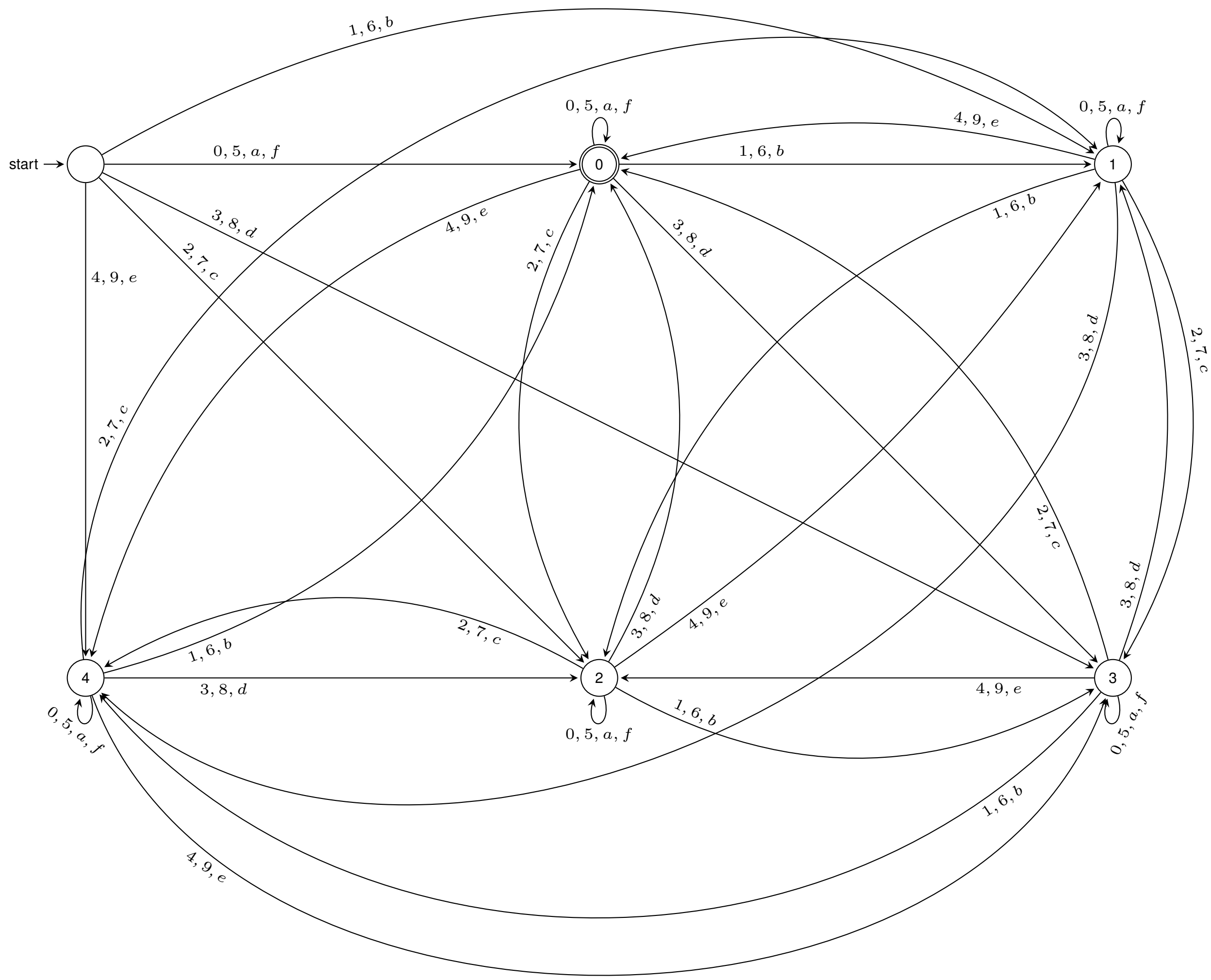
3. [10 marks] Recognizing hex integers divisible by 5

Let string $s \in \{0 - 9, A - F\}^*$. Let n be string s interpreted as a hexidecimal integer. Draw a DFA that accepts s if and only if:

$$n \equiv 0 \mod 5.$$

Assume $\varepsilon \not\equiv 0 \mod 5$.

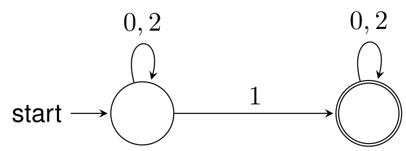
Answer: We need 6 states: one state for each possible outcome of a number $\mod 5$, plus a state to handle the empty string. The next is to add transitions based on the next character. Numerically if you take a hexidecimal number n and tack digit d on the end, the result will be $16 \cdot n + d$. Notice that $16 \cdot n + d \equiv n + d \mod 5$. So whatever state you're in, *plus* whever the new character is, mod 5, that's your next state. So now you just need to draw all the transitions...



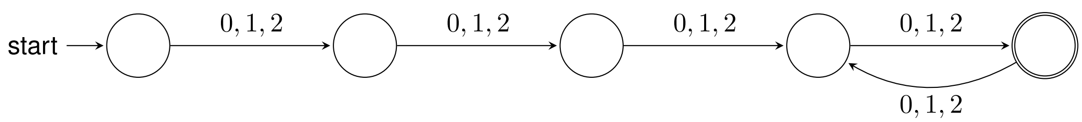
4. [6 marks] Design NFAs

Let $\Sigma = \{0, 1, 2\}$. Draw an NFA recognizing each of the following languages:

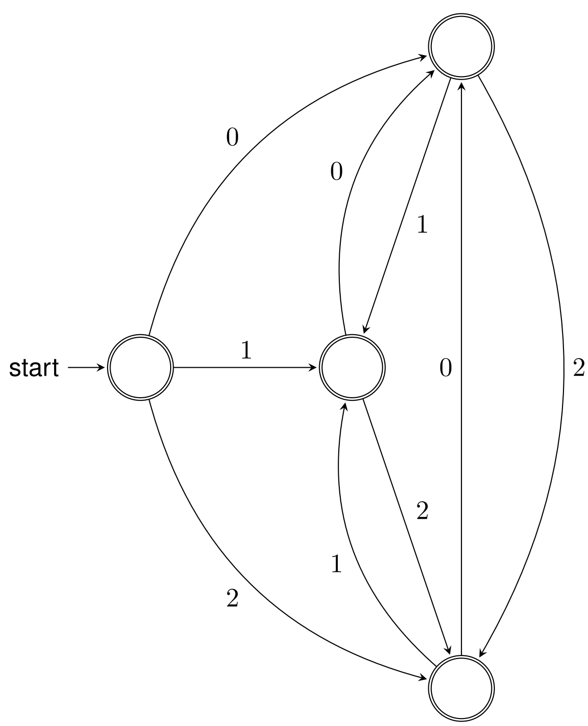
(a) [2 marks] The set of strings that contain a single 1,
Answer:



(b) [2 marks] The set of strings with an even length n where $n > 2$
Answer:



(c) [2 marks] The set of strings that contain no consecutive digits (i.e., a 0 cannot follow a 0, etc).
Answer:



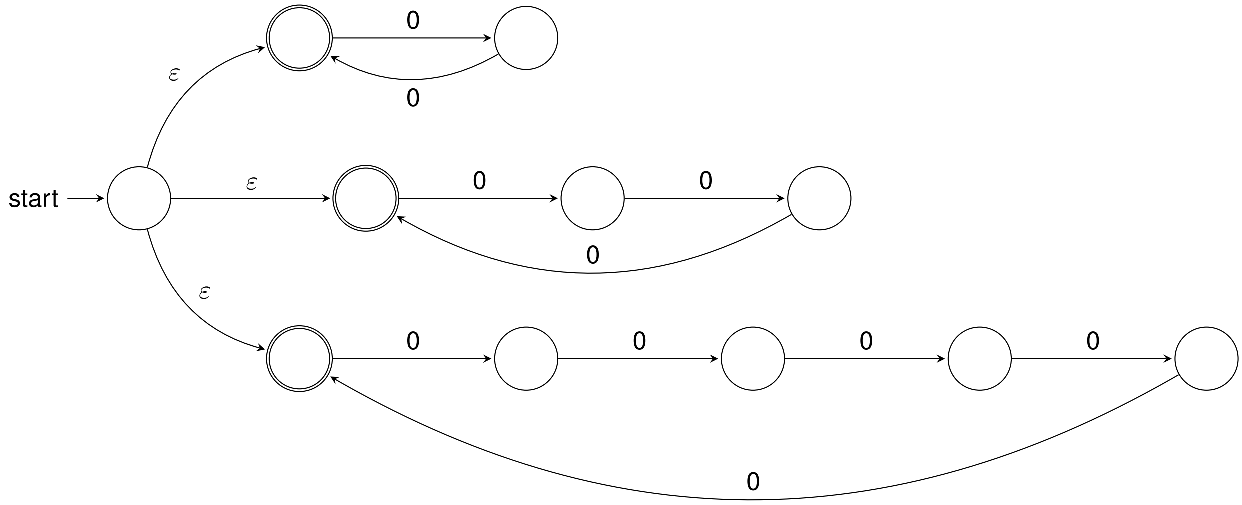
5. [10 marks] An NFA in an Economy of States

Let $s \in \Sigma = \{a\}^*$. Let $|s|$ denote the length of string s. Construct a finite automaton in less than a *dozen* states that recognizes language:

$$L = \{s : \text{gcd}(|s|, 300) \neq 1\},$$

where gcd(a, b) denotes the greatest common divisor between a, b.

Answer: The first thing to notice is that $300 = 2^2 * 3 * 5^2$. The next is to notice that the gcd$(300, s) \neq 1$ if s is a multiple of 2, 3, or 5. So the machine should accept if the input length is $0 \bmod 2, 0 \bmod 3$, or $0 \bmod 5$. We can use ϵ transitions to handle these three cases separately:

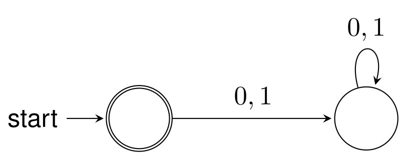


6. [10 marks] Prove Finite Languages are Regular

We say a language L is *finite* if L contains a finite number of strings. Using induction, prove all finite languages are regular.

Answer: The solution is as follows:
Without loss of generality, assume $\Sigma = \{0, 1\}$. The proof proceeds similarly for langauges defined over any other alphabet.

- **Basis step**. Show the statement holds for n=0. Let $L_0 = \phi$ be a language containing $n = 0$ strings. Show L_0 is regular. To do this, it is sufficient to give a DFA that recognizes L_0:



- **Inductive step**. Let L_n be a finite regular language for which $|L_n| = n$ for some $n \geq 0$. We must show L_{n+1} is also regular. Suppose $L_{n+1} = L_n \cup \{s\}$ for some string s not contained in L_n. Then if $|L_n| = n$ then $|L_{n+1}| = n + 1$. As we proved in class, regular languages are closed under union, meaning if L_n and $\{s\}$ are regular, then so is $L_{n+1} = L_n \cup \{s\}$.

First we note L_n is regular by definition. Ok, so what about $\{s\}$? Well this is a language that contains a single string, s. By the definition of formal languages, s has finite length. Suppose $s = s_1 s_2 \ldots s_{k-1} s_k$ for $k > 0$ and $s_i \in \{0, 1\}$. Then an NFA recognizing s can be constructed in $k + 1$ states (as per our lecture on string matching). Finally, L_{n+1} can be constructed as the union:

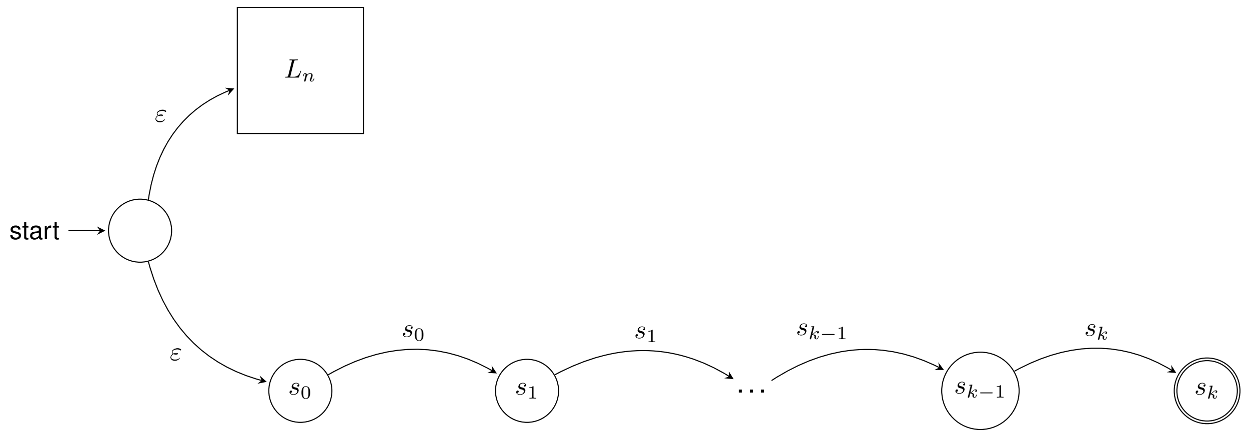


Figure 1: Figure for $L_{n+1} = L_n \cup \{s\}$.

Clearly if L_n is regular, L_{n+1} is also regular. That is, if a language containing n strings is regular, then a language containing $n + 1$ strings is also regular as per the inductive step. And we have L_0 is regular as per the basis step. Therefore L_{n+1} is regular for all natural numbers n, and thus any finite language is regular. □