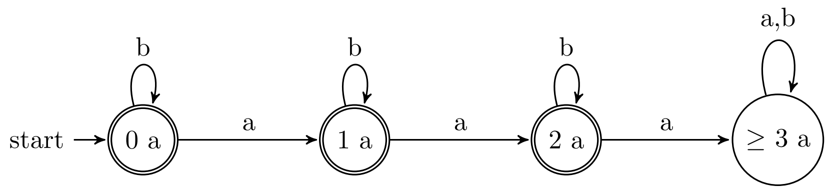
CS 164	**Fall 2015**

WA1

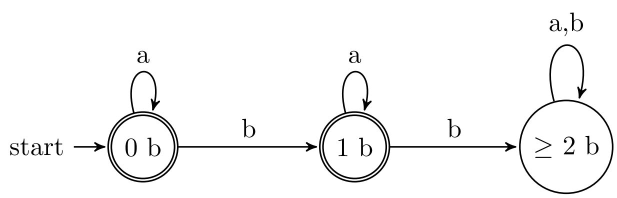
Nikhil Unni (cs164-es), Section : Monday 3pm

1. Give a DFA for the following languages over the alphabet $\Sigma = \{a, b\}$:

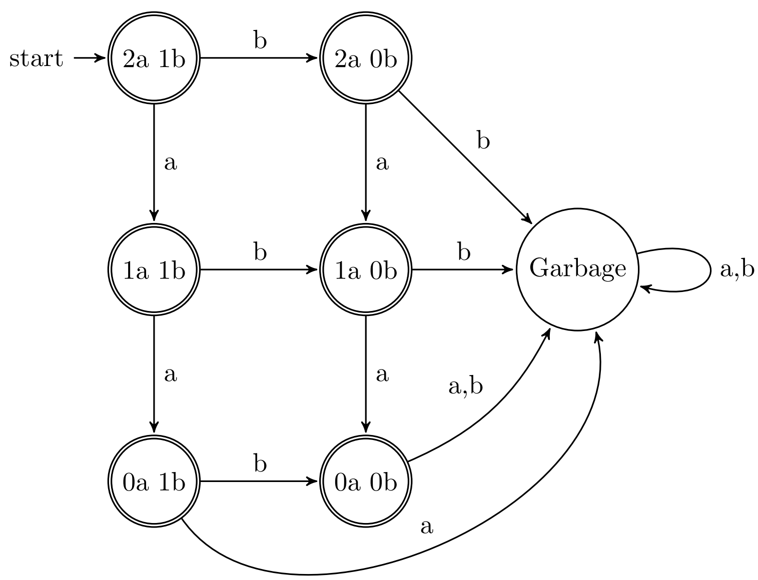
   - All strings that contain at most two a's.



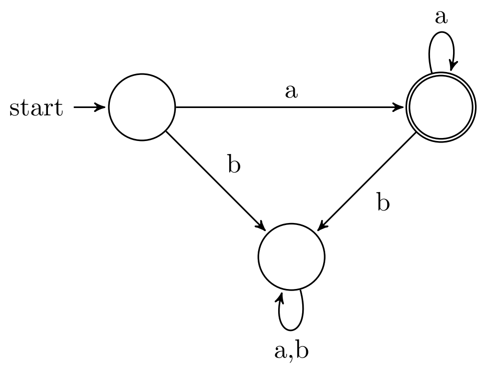
   - All strings that contain at most one b.



   - All strings that contain at most two a's and at most one b.



   - All strings that contain at least one a and no b's.

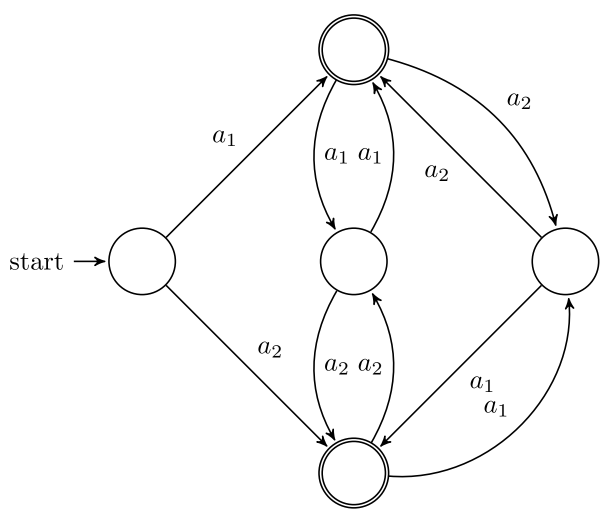


2. Consider the following DFA over the alphabet $\Sigma = \{a, b\}$. (Not pictured.)
   Give a one sentence description of the language recognized by the DFA. Write a regular expression for the same language.

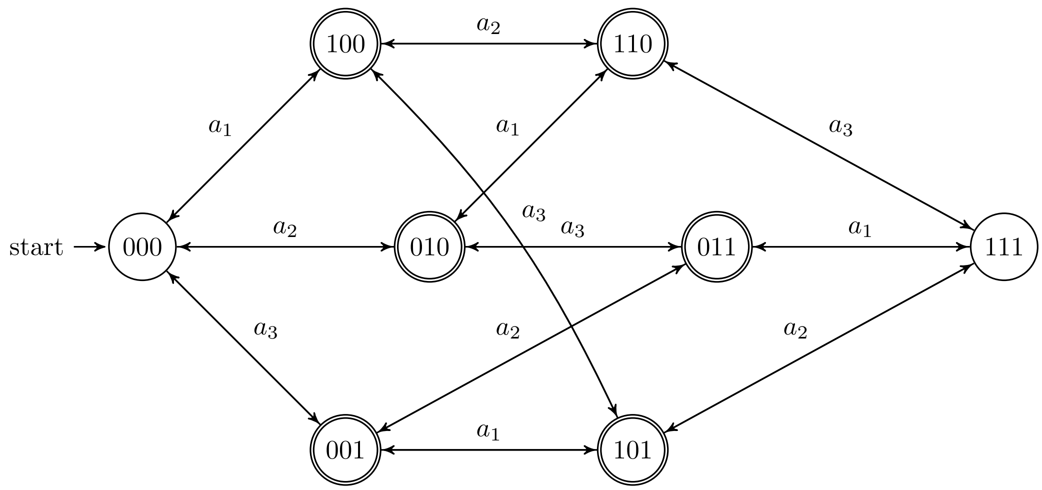
   It is the set of words where the number of a's is divisible by 4. It can be represented by the regex $b^*(ab^*ab^*ab^*ab^*)^*$.

3. Let $\Sigma_m = \{a_1, \cdots, a_m\}$ be an alphabet containing m elements, for some integer $m \geq 2$. Let L_m be the following language that includes all strings in which at least one of the characters occurs an odd number of times and one of the characters occurs an even number of times.

   Construct a DFA for the language L_2.



   Also construct an NFA for the language L_3.



In the diagram, the nodes are labeled "001" or "011" for instance. This just means that, at index i, if it is a 0, that means the number of a_{i+1} is even, or its odd if the character is 1. Also, another note – I drew double arrows for the transitions, which officially means : if there's a double arrow with character a between two nodes q_0 and q_1, you can transition from q_0 to q_1 with a, and you can transition from q_1 to q_0 with a. I only did it this way because without the double arrows, an already hard to read graph became an impossible to read graph.

4. (a) Determine whether or not the following languages are regular. Explain why in one or two sentences.

   - L_1: All strings over the alphabet $\{0, 1\}$ that have a different number of 1's and 0's.

     Not regular, because there's no **finite** "mechanism" for keeping track of the differences in 0's and 1's that we've seen so far.

   - L_2: All strings over the alphabet $\{0, 1\}$ that are not repeating sequences.

     Not regular, because there's no way to "remember" the (arbitrary amount of) nodes we've seen before. It can by shown by the Pumping Lemma that since the sequence we have to remember can be of arbitrarily long length, there's no way that there can be a DFA that accepts the language.

   - L_3: All words that have ever existed and will ever exist in the English language (Assume there is a maximum word ength since humans have limited breath).

     Because there is a maximum word length, call it n characters long, the total number of words is a finite 26^n (or some number greater than 26 if humans develop new characters... but certainly a finite alphabet length). And since all finite languages are regular (we can construct a DFA/Trie for the set of all possible words), L_3 is regular.