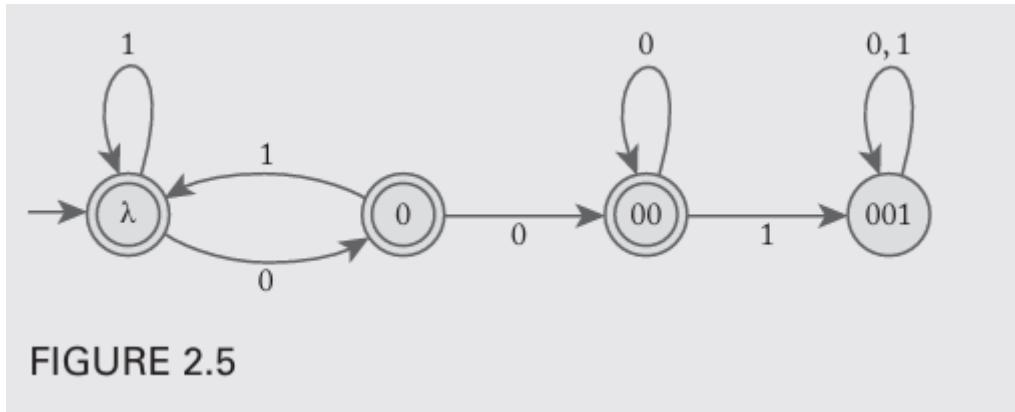


A) Complete problem 2 on page 51. Specify the full DFA $L(M) = (Q, \Sigma, \delta, q_0, F)$.

The problem: Translate the graph in Figure 2.5 into δ -notation.



Answer.

$$Q = \{\lambda, 0, 00, 0001\}$$

$$\Sigma = \{0, 1\}$$

$$q_0 = \lambda$$

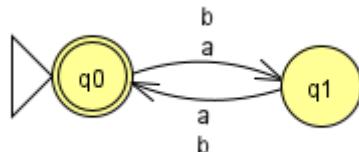
$$F = \{\lambda, 0, 00\}$$

$$\begin{aligned} \delta = & \{(\lambda, 0) \rightarrow 0, (\lambda, 1) \rightarrow \lambda, (0, 0) \rightarrow 00, (0, 1) \rightarrow \lambda, \\ & (00, 0) \rightarrow 00, (00, 1) \rightarrow 001, (001, 0) \rightarrow 001, (001, 1) \rightarrow 001\} \end{aligned}$$

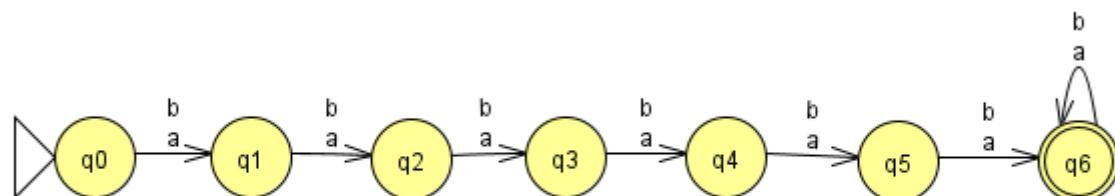
B) Complete problems 3 and 5 on page 51. Note that when the problem says “DFA” in problems 3 and 5 it means the transition graph.

3. For $\Sigma = \{a, b\}$, construct dfa's that accept the sets consisting of

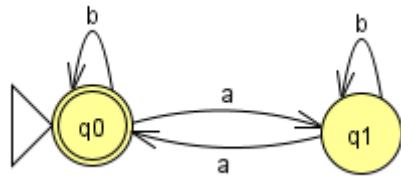
(a) all strings of even length.



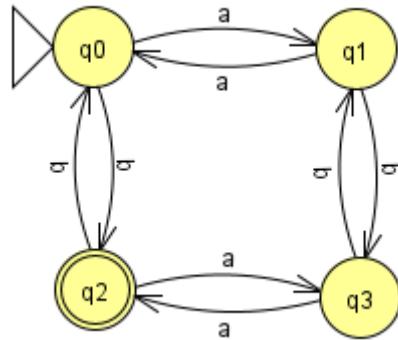
(b) all strings of length greater than 5.



(c) all strings with an even number of a's.

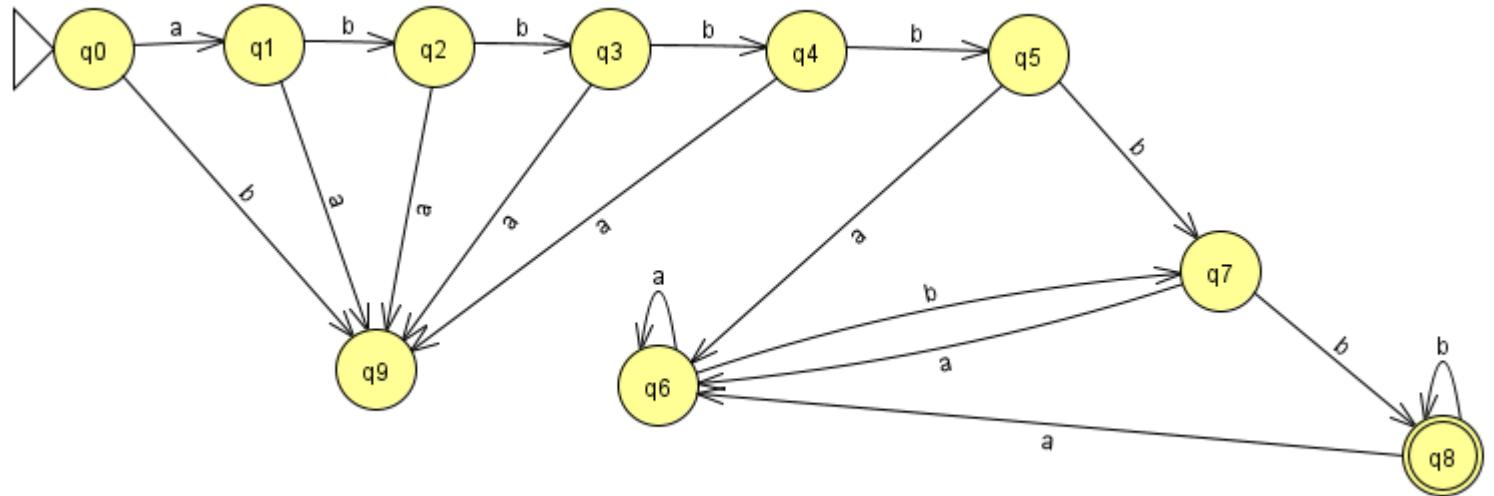


(d) all strings with an even number of a's and an odd number of b's.

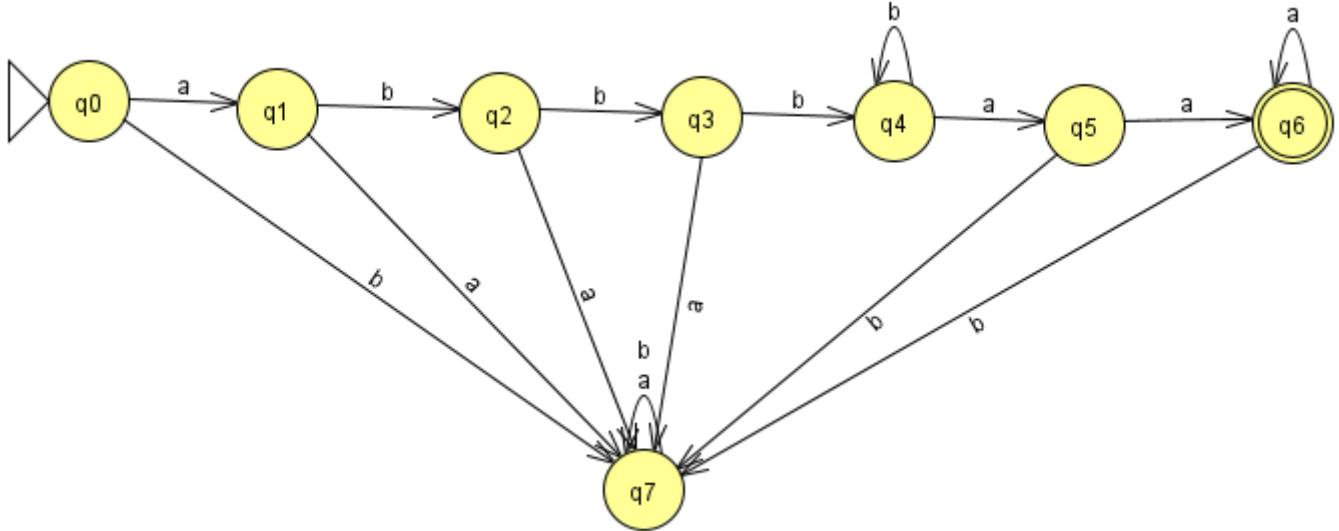


5. Give dfa's for the languages

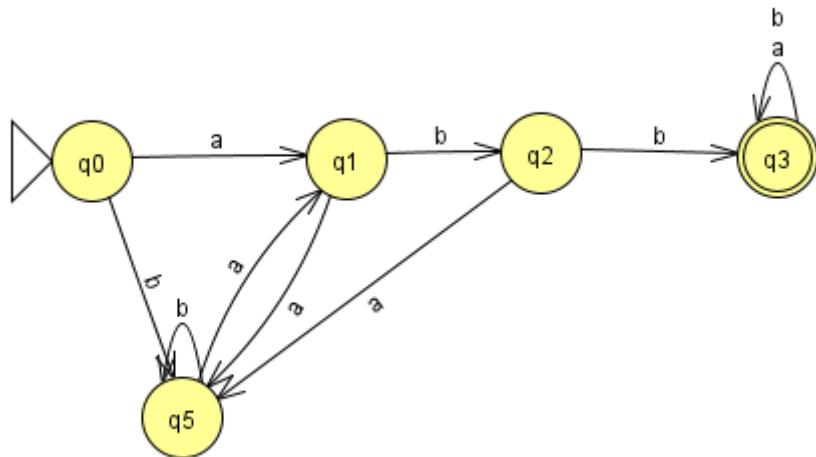
(a) $L = \{ab^4wb^2 : w \in \{a, b\}^*\}$.



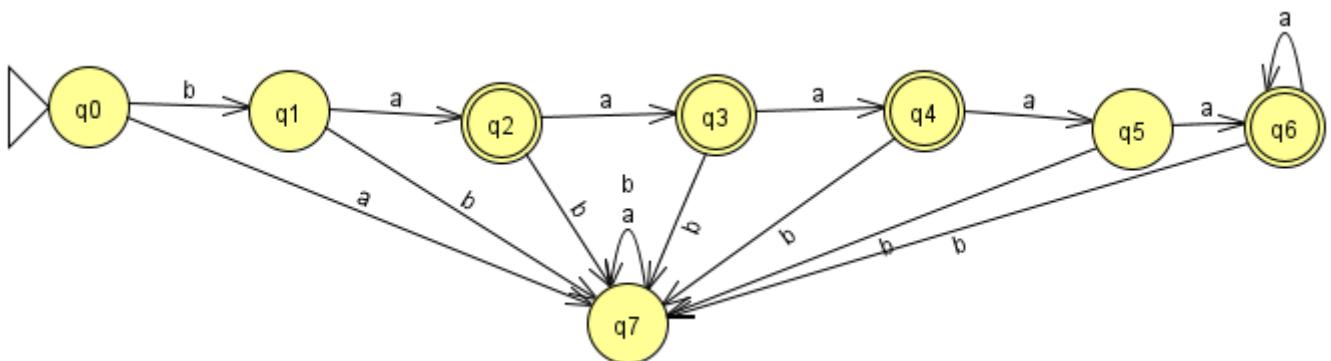
(b) $L = \{ab^n a^m : n \geq 3, m \geq 2\}.$



(c) $L = \{w_1abbw_2 : w_1 \in \{a, b\}^*, w_2 \in \{a, b\}^*\}.$



(d) $L = \{ba^n : n \geq 1, n \neq 4\}.$



- C) Implement your DFA from problem #5 (a) as a C++ program. The program should accept any string from the user and print “accept” if the string is in the language and “reject” if the string is not in the language. You may **not** use a regular expressions library. Submit a print out of your source code.

```
1 #include <iostream>
1 #include <string>
2 using namespace std;
3
4 /* Name: Lucas Hasting
5 * Date: 9/5/2025
6 * Class: CS 421
7 * Instructor: James Jerkins
8 * Sources: https://cplusplus.com/reference/string/string/
9 */
10
11 int main(){
12     //used to store input string
13     string input;
14
15     //used to determine if middle contains alphabet
16     bool middle = true;
17
18     //get user input
19     cin >> input;
20
21     //if the string is not of min size, reject
22     if(input.size() < 6)
23         cout << "reject" << endl;
24
25     //if string starts with abbbb, continue, otherwise reject
26     else if(input.substr(0, 5) == "abbbb"){
27         //determine if middle contains alphabet
28         for(int i = 5; i < input.size() - 2; i++){
29             middle &= (input[i] == 'a' || input[i] == 'b');
30         }
31
32         //if string ends with bb (any word in the middle), accept, reject otherwise (also if no alpha from L in middle)
33         if(middle & input.substr(input.size()-2, input.size()) == "bb")
34             cout << "accept" << endl;
35         else
36             cout << "reject" << endl;
37     } else
38         cout << "reject" << endl;
39
40     return 0;
41 }
```