1. List the elements of $\{a, ab\}^*$. Find a way to describe strings in this language, i.e., find a predicate $P(s)$ such that $P(s)$ is true iff s in $\{a, ab\}^*$, for all strings s over alphabet $\{a, b\}$.

Ans: Let's first list the elements of $\{a, ab\}^*$:

$$\begin{aligned}\{a, ab\}^* &= \{e\} \cup \{a, ab\} \cup \{aa, aab, aba, abab\} \cup \{aaa, \cdots\} \cup \cdots \\ &= \{e, a, ab, aa, aab, aba, abab, aaa, aaab, aaba, aabab, \cdots\}\end{aligned}$$

This is almost like $\{a, b\}^*$ (all strings of a's and b's), except each b comes with an a in front, i.e., $P(s) = s$ is a string of a's and b's where each b is immediately preceded by an a. Are we done? NO! we should check that $s \in \{a, ab\}^*$ iff $P(s)$, by arguing each direction:

$\Rightarrow$*:* It is obvious that each string in $\{a, ab\}^*$ has property $P()$ (every b immediately preceded by an a).

$\Leftarrow$*:* Moreover, each string where each b is immediately preceded by an a can be broken up into pieces ab (one for each b in the string) with every other symbol being a. Hence, the string belongs to $\{a, ab\}^*$. □

2. Find three different examples of a language L over alphabet $\{a, b, c\}$ such that $L = L^{\circledast}$.

Ans:

- $L = \{\epsilon\}$: $L^{\circledast} = \{\epsilon\} = L$
- $L = \{\epsilon, a, aa, aaa, \cdots\} = \{a\}^*$: as above, $L^{\circledast} = L$
- $L = \{a, b, c\}^* = \{\epsilon, a, b, c, aa, ab, ac, ba, bb, bc, ca, cb, cc, \cdots\}$:
  - $s \in L^{\circledast}$ iff $s = s_1 \circ s_2 \circ \cdots \circ s_k$ for some $s_1, \cdots, s_k \in L$. Each $s_i \in L$ is a sequence of a's, b's, c's, and hence s is a sequence of a's, b's, and c's. So $s \in L$.
  - $s \in L$ iff $s = s_1 \circ s_2 \circ \cdots \circ s_k$ for some $s_1, \cdots, s_k \in \{a, b, c\}$ – but then, $s \in L^{\circledast}$ because $\{a, b, c\} \subseteq L^{\circledast}$.

□

3. Give a DFA for each language below.

a) $L_1 = \{s \in \{0, 1\}^* :$ s contains at least 2 characters and s's second character is a 1 $\}$

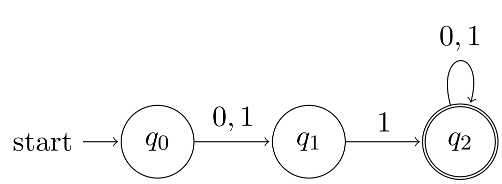
b) $L_2 = \{s \in \{0, 1\}^* :$ s contains fewer than 2 characters$\}$

c) $L_3 = \{s \in \{a, b\}^* :$ every a in s is eventually followed by b$\}$. For example, $aaab \in L_3$ because there is a b that follows every a– even though it is not immediately after the first two as.

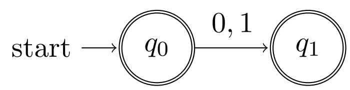
d) $L_4 = \{s \in \{a, b\}^* :$ the third-last character of s is a b $\}$

*Ans:*Considering the conventions that we used to draw DFAs, we have:

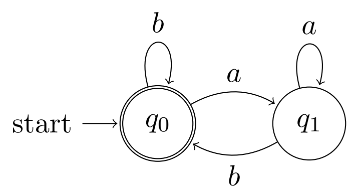
a)



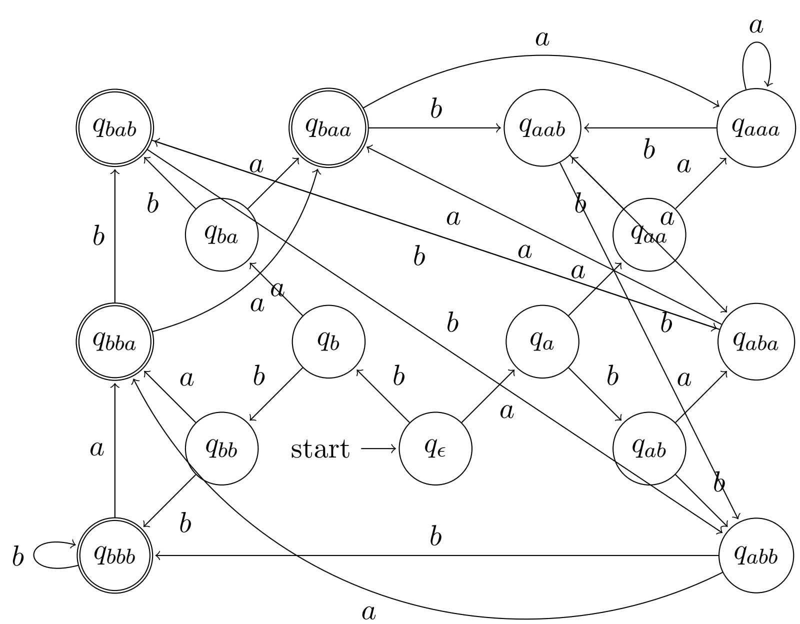
b) Note that the description suggests $L_2 = \{\epsilon, 0, 1\}$. Hence,



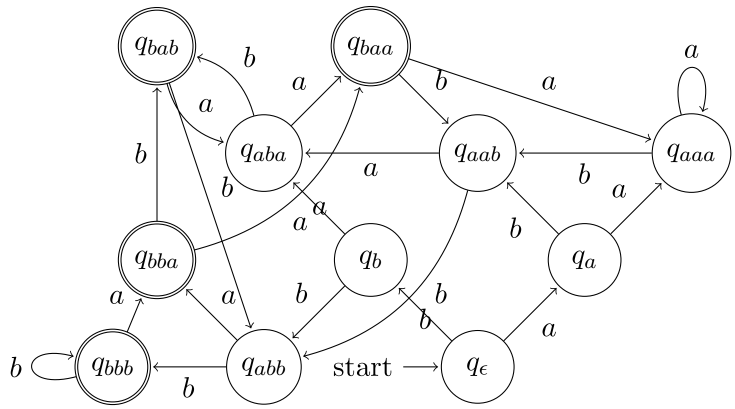
c) From the definition of L_3, we can conclude that string $s \in L_3$ cannot end with an a. We can use this to find:



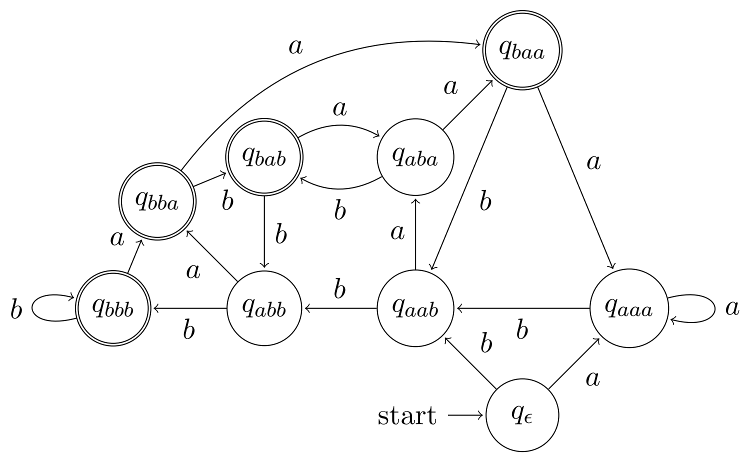
d) We need information about the 3rd last character. Because of that, we need to keep track of the last three characters that we've seen. The straightforward way is to label states by all the possible strings of length ≤ 3. This way, we end up with 15 states: $\{q_\epsilon, q_a, q_b, q_{aa}, q_{ab}, q_{ba}, q_{bb}, q_{aaa}, q_{aab}, q_{aba}, q_{abb}, q_{baa}, q_{bab}, q_{bba}, q_{bbb}\}$. q_ϵ is the initial state (representing that the string so far is empty) and each state q_{bxy} is accepting (representing that the last three characters are "bxy"); each other state is rejecting.



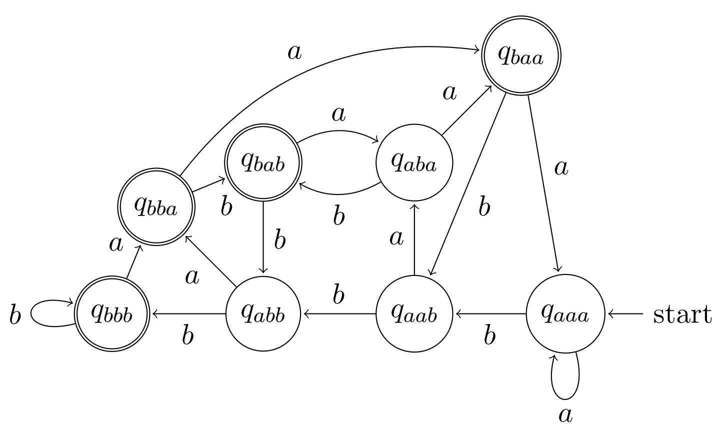
As you can see this graph is really complicated and hard to draw. There is a trick to simplify this a bit. Notice that the transitions from states q_{xy} are identical to the transitions from states q_{axy}, and these states are all rejecting. So we can remove the states q_{xy} and replace transitions going into them with transitions going into q_{axy}.



Similarly, states q_x behave exactly the same as states q_{aax}.



Finally, state q_ϵ behaves exactly the same as state q_{aaa}.



□