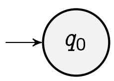
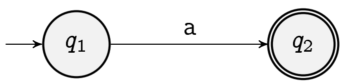
CSC236 tutorial exercises, Week #11
sample solutions

1. Recall that $\emptyset$ is a valid (but seldom used) symbol in our regular expression syntax. Construct an NFSA M for the regular expression $R = \emptyset a$, using the procedure described in section 7.6.1 of the course notes. Does the result make sense with the meaning of R?

   Solution:
   We begin by constructing separate FSAs for the sub-expressions $\emptyset$ and a. The following FSA $M_1 = (Q_1, \Sigma, \delta_1, s_1, F_1)$ is equivalent to $\emptyset$:

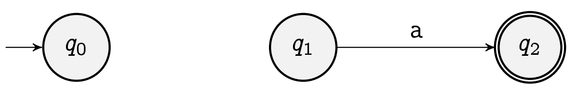
   

   The following FSA $M_2 = (Q_2, \Sigma, \delta_2, s_2, F_2)$ is equivalent to the RE a:

   (Both of these are given by figure 7.20 in the course notes.)

   To construct an NFSA that accepts the concatenation of these languages, we use the construction from figure 7.18 of the course notes. This entails connecting all the accepting states F_1 of M_1 with s_2, the start state of M_2, via epsilon transitions, and making them non-accepting in the new FSA. However, in this case $F_1 = \{\}$, so our new FSA M is just:

   This may look weird, but no part of our formal definition of FSAs forbids the existence of a state machine like this, where we have an 'island' of states not reachable from the starting state.

   It is clear by inspection that $\mathcal{L}(M) = \emptyset$, since no string can reach the accepting state q_2. And this agrees with the meaning of the RE $\emptyset a$. Recall that, in general, when concatenating two regular expressions S and T, the resulting language is the concatenation of the languages of those REs, i.e. $\mathcal{L}(ST) = \mathcal{L}(S)\mathcal{L}(T) = \{st \in \Sigma^* \mid s \in \mathcal{L}(S), t \in \mathcal{L}(T)\}$.

   Since $\mathcal{L}(\emptyset)$ is the empty language, $\mathcal{L}(\emptyset)\mathcal{L}(a)$ is also empty, agreeing with our FSA.

2. Several of the closure results from this week's readings become a bit simpler to prove if we assume that our FSA has only one accepting state. In this question, you will prove that this is something we can assume without loss of generality.

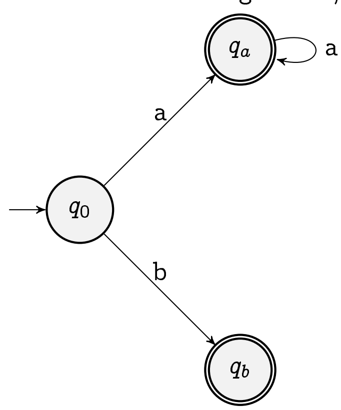
(a) Show that for any FSA M, there exists an NFSA M' having only one accepting state, such that $\mathcal{L}(M) = \mathcal{L}(M')$. (You may use graphics to illustrate your construction, as in the Vassos notes, or describe it in words.)

Solution:
Let $M = (Q, \Sigma, \delta, s, F)$ be an arbitrary DFSA (we can assume determinism without loss of generality, because we know that any NFSA can be converted to a DFSA using the subset construction). We construct an equivalent NFSA M' with a single accepting state. M' will be identical to M except for the following changes:

- We add a new state q_f
- q_f is the only accepting state
- For each previously accepting state $q \in F$, we add an ε-transition from q to q_f.

(b) Is it also true that there always exists an equivalent *D*FSA with only one accepting state? If yes, give a constructive procedure for creating such a DFSA. If not, find a counterexample (and try to argue why it has no equivalent DFSA with a single accepting state).

Solution:
Consider the following DFSA, equivalent to the RE $(b + a^*)$:

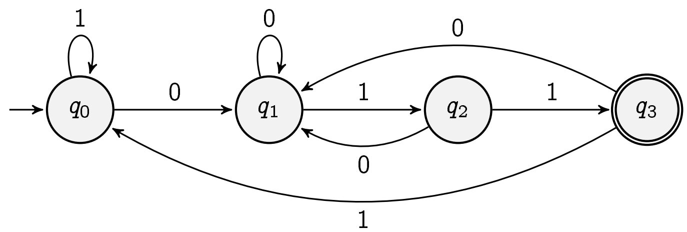


Suppose for contradiction there exists an equivalent DFSA $M = (Q, \Sigma, \delta, s, \{q_f\})$ with a single accepting state, q_f. It follows that $\delta^*(s, a) = \delta^*(s, b) = q_f$, since the strings a and b must both be accepted. What is $\delta(q_f, a)$? If it is q_f, then we inappropriately accept the string ba. If it's a non-accepting state, then we inappropriately reject the string aa. Either way, M is not equivalent to our original DFSA.

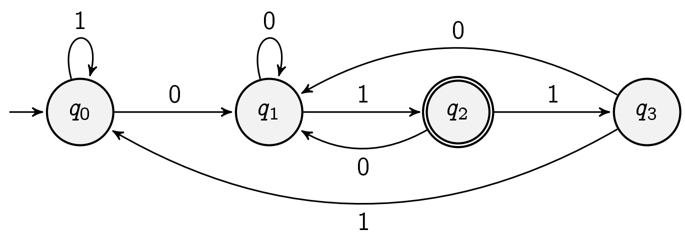
This argument uses the concept of *indistinguishable prefixes*, which will be introduced in more detail next week.

3. For a language L, define $\text{Trim}(L)$ to be the language resulting from truncating the last symbol from each string in L. Formally $\text{Trim}(L) = \{x \in \Sigma^* \mid \exists w \in L, a \in \Sigma, w = xa\}$. In this question, you will show that the set of languages recognized by FSAs is closed under 'trimming'.

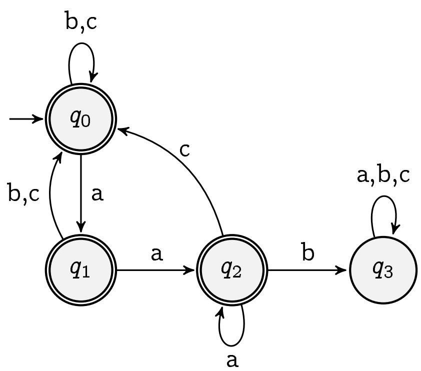
(a) Modify the following DFSA M to form a new FSA M' such that $\mathcal{L}(M') = \text{Trim}(\mathcal{L}(M))$. (You may use non-determinism if you wish.)



Solution:
$\mathcal{L}(M)$ is the language of strings ending in 011 (you may recognize this FSA from last week's quiz), so $\text{Trim}(\mathcal{L}(M))$ is the set of strings ending in 01. The only necessary change is flipping the accepting state from q_3 to q_2:



(b) Repeat the procedure above for the following DFSA:



Solution:
This DFSA accepts all strings in $\{a, b, c\}^*$ not containing the substring aab. The result of trimming this language is not so obvious. In fact, it turns out that $\text{Trim}(\mathcal{L}(M)) = \mathcal{L}(M)$, so we need to make no modifications to the DFSA in this case.

There are two ways to convince ourselves of this. We can think in terms of the mutual inclusion of the languages. If we start with a string not containing aab and truncate its last symbol, then the result clearly also doesn't contain aab, so $\text{Trim}(\mathcal{L}(M)) \subseteq \mathcal{L}(M)$. But it's also the case that for any string not containing aab, that string is also an element of the trimmed language, because we can always append an a to get another string in $\mathcal{L}(M)$, so $\mathcal{L}(M) \subseteq \text{Trim}(\mathcal{L}(M))$. Thus $\text{Trim}(\mathcal{L}(M)) = \mathcal{L}(M)$.

We can also argue this based on the operation of the DFSA. By inspection, a string that reaches any of the accepting states can also reach an accepting state by following one additional transition.

(c) Describe a general procedure for constructing an FSA that accepts $\text{Trim}(\mathcal{L}(M))$ given an arbitrary DFSA M.

Solution:
Our derived FSA M' will be identical to M except that a state will be accepting in M' if and only if there is a transition from that state to one of M's accepting states. In other words, letting F and F' be the accepting states of M and M', respectively, $q \in F' \iff \exists a \in \Sigma, q_f \in F, \delta(q, a) = q_f$.