1. Extra Problems for Lecture 14

a. In the context of natural languages, what is the type (set, string, number, etc.) and meaning of each of these symbols?

$$\Sigma \qquad \Sigma^* \qquad \varepsilon$$

Σ is an alphabet, or a set of symbols (single characters)

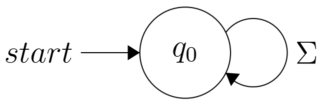
Σ^* is a set of languages made of symbols from Σ. A language is a set of strings.

ε is a string with no characters in it, i.e. the empty string.

b. Let $\Sigma = \{a, b\}$. Are each of these automata DFAs over Σ? Why or why not? If it is a DFA, what language does it accept? (Hint: It's a good idea to test out some strings and see which ones work.)

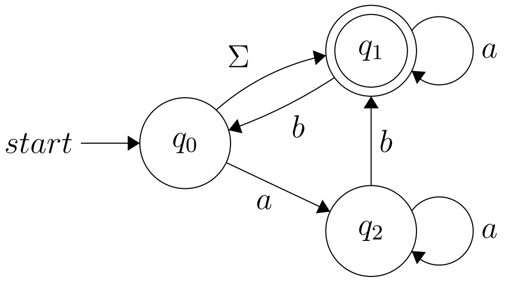
Here are the rules for DFAs, as a refresher: For each state in the DFA, there must be exactly one transition defined for each symbol in Σ; there is a unique start state; and there are zero or more accepting states.

(1)



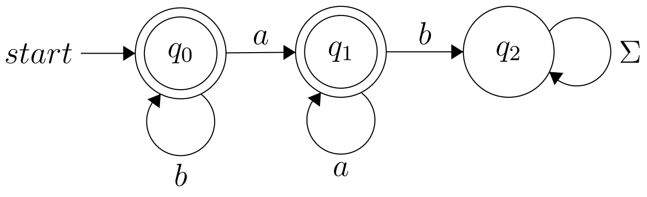
This is a DFA and accepts $\varnothing$.

(2)

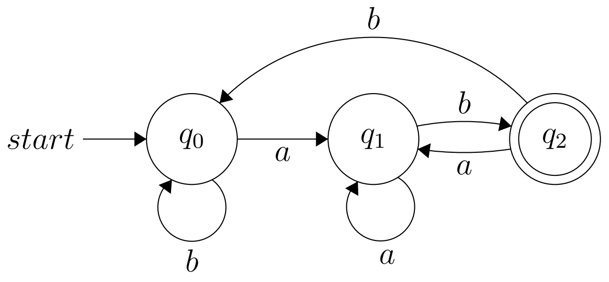


This is NOT a DFA because q_0 has two transitions defined for the symbol a, since a is included in Σ.

(3)



This is a DFA and accepts the set of all strings that don't contain the substring *ab*.



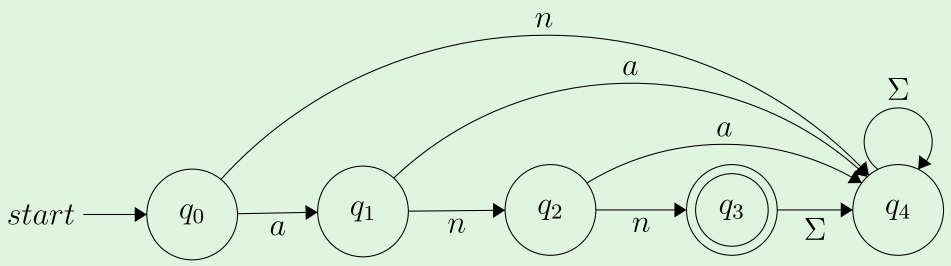
(4)

This is a DFA and accepts the set of all strings that end in *ab*.

2. Extra Problems for Lecture 15

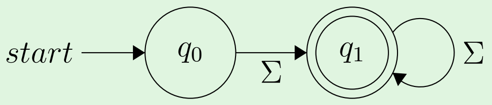
a. Let $\Sigma = \{n, a\}$. Design a DFA that recognizes each of these languages:

(1) $\{\texttt{ann}\}$

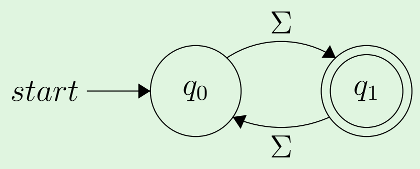


(2) $\bar{L}$, given that $L = \{\varepsilon\}$ (Recall that the notation $\bar{L}$ refers to the complement of the language L, meaning the language of all strings in Σ^* that are not in L.)

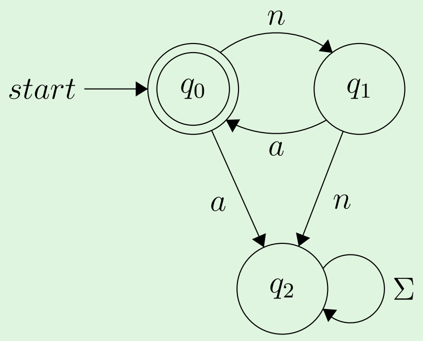
One way to arrive at this solution is to draw a DFA for L and then flip the accept/reject states.



(3) $\{w \in \Sigma^* \mid w \text{ has an odd length }\}$

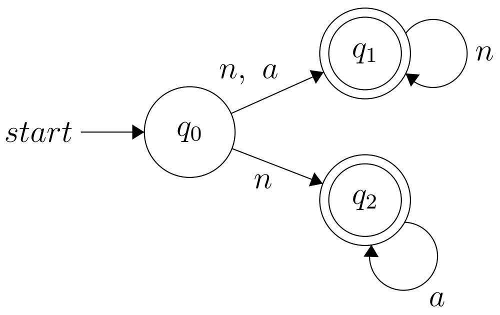


(4) $\{\varepsilon, \texttt{na}, \texttt{nana}, \texttt{nanana}, ...\}$



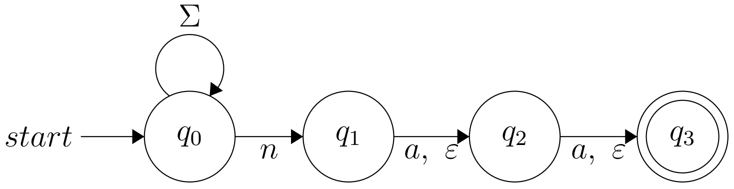
b. Let $\Sigma = \{n, a\}$. Given the following NFAs over Σ, determine which language the NFA recognizes. It's okay to describe the language with set-builder notation or with words. Recall that an NFA accepts if any set of choices results in the automaton ending up in an accept state with the entire input having been read through.

(1)

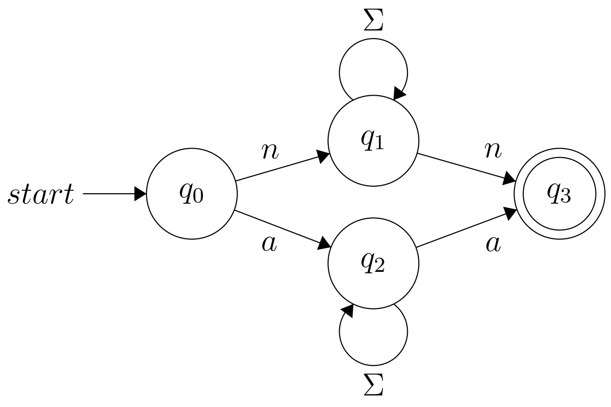


This NFA recognizes the language of non-empty strings containing all n's, strings consisting of one a followed by any number of n's, and strings consisting of one n followed by any number of a's.

(2)



This NFA recognizes the language of strings that end in n, na, or naa.

(3) 

This NFA recognizes the language of strings whose first and last letter is the same, not including strings that only have one letter in them.

3. Extra Problems for Lecture 16

a. Read the Guide to the Subset Construction on the website. Then, use the subset construction to convert the DFA from 2b(3) into an NFA.

State	a	n
$\{q_0\}$	$\{q_0\}$	$\{q_0, q_1, q_2, q_3\}$
$^*\{q_0, q_1, q_2, q_3\}$	$\{q_0, q_2, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$^*\{q_0, q_2, q_3\}$	$\{q_0, q_3\}$	$\{q_0, q_1, q_2, q_3\}$
$^*\{q_0, q_3\}$	$\{q_0\}$	$\{q_0, q_1, q_2, q_3\}$

b. What is the relationship between regular languages and NFAs/DFAs?

Any regular language can be recognized by an NFA/DFA, and any language recognized by an NFA/DFA is regular.

c. If L_1 and L_2 are languages over Σ, describe the language $L_1 - L_2$ in words. Is this language regular? Why or why not? (Hint: Can you express it in terms of language operations on languages you know are regular?)

This is the language of strings that are in L_1 but not in L_2.

One way to see why this language is regular is to observe that it is equal to $L_1 \cap \overline{L_2}$, and regular languages are closed under complement and intersection.