Finite Automata are Closed Under Regular Operations
Theory of Computation (CSCI 3500)

Prof. Harley Eades (heades@gru.edu).

Read chapter 1.2.

First, lets recall the definition of an NFA:

Definition 1. *A **non-deterministic finite automata (NFA)** is a 5-tuple,* $(Q, \Sigma, \delta, q_0, F)$ *where:*

Q *is the set of states of the NFA,*
Σ *is the alphabet,*
$\delta : Q \times (\Sigma \cup \{\epsilon\}) \to \mathcal{P}(Q)$ *is the transition function,*
q_0 *is the start state, and*
F *is the set of final states.*

Since we know that NFAs are equivalent to DFAs, and thus, the languages NFAs recognize are indeed regular languages, then in this lecture we will only need to build NFAs. This point will make our proofs a lot simpler.

We now show that finite automata are closed under a set of operations called regular operators. These operators allow one to construct larger finite automata from one or more smaller automata. There are four regular operators:

- Union
- Complement
- Intersection
- Kleene star

We cover each of these in the following sections.

1 Closure under Union

In this section we show that given two NFAs M_1 and M_2 we can construct a new NFA denoted $M_1 \cup M_2$ called the union of the NFAs M_1 and M_2.

Suppose $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ are NFAs. Then we construct their union $M_1 \cup M_2 = (Q_\cup, \Sigma_\cup, \delta_\cup, q_\cup, F_\cup)$ as follows:

$$Q_\cup = \{q_0\} \cup Q_1 \cup Q_2$$
$$\Sigma_\cup = \Sigma_1 \cup \Sigma_2$$

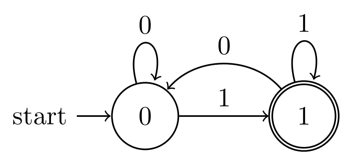
$$\delta_\cup(q,a) = \begin{cases} \{q_1, q_2\} & : q = q_0 \text{ and } a = \epsilon \\ \emptyset & : q = q_0 \text{ and } a \neq \epsilon \\ \delta_1(q,a) & : q \in Q_1 \text{ and } a \in \Sigma_1 \\ \delta_1(q,a) & : q \in Q_1 \text{ and } a = \epsilon \\ \delta_2(q,a) & : q \in Q_2 \text{ and } a \in \Sigma_2 \\ \delta_2(q,a) & : q \in Q_2 \text{ and } a = \epsilon \end{cases}$$

$$q_\cup = q_0$$
$$F_\cup = F_1 \cup F_2$$

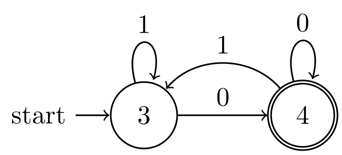
Note that q_0 is a new start state such that $q_0 \notin Q_1 \cup Q_2$. It is clear that this new NFA accepts the language $L(M_1) \cup L(M_2)$. The previous construction requires that $Q_1 \cap Q_2 = \emptyset$, however, this can always be achieved by a simple renaming of the labels on states. With out loss of generality we will assume that distinct finite automata have distinct sets of states throughout this entire note.

We now consider an example. Suppose we have the following two NFAs:

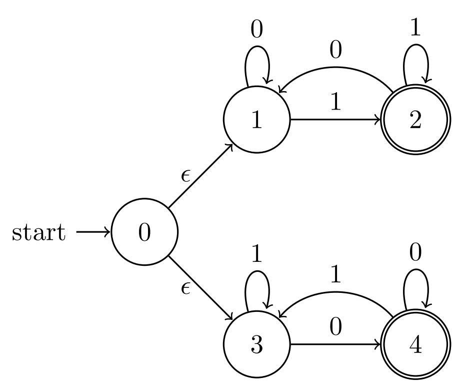
M_{odd}:



M_{even}:



$M_{\text{odd}} \cup M_{\text{even}}$:



The previous construction amounts to a proof of the following lemma:

Lemma 2 (Union of two Regular Languages)**.** *The class of regular languages is closed under union.*

Proof. Suppose L_1 and L_2 are regular. Then it suffices to show that $L_1 \cup L_2$ is a regular language. By regularity we know that there must exist two NFAs M_1 and M_2 such that $L(M_1) = L_1$ and $L(M_2) = L_2$.

Using the previously given construct we can construct a NFA $M_1 \cup M_2$ such that $L(M_1 \cup M_2) = L(M_1) \cup L(M_2)$. Thus, by the NFA-NFA equivalence we know $L(M_1) \cup L(M_2) = L_1 \cup L_2$ is regular. $\square$

2 Closure under Complement

We now show that given a NFA, M, we can construct a new NFA denoted $\overline{M}$, such that, $L(\overline{M}) = \overline{L(M)}$.

Lemma 3 (Complement of a Regular Language)**.** *The class of regular languages is closed under complement.*

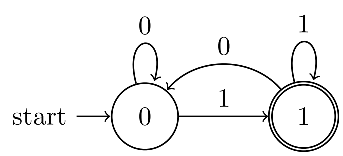
Proof. Suppose L is a regular language. Then it suffices to show that that the complement of L, $\overline{L}$, is regular. Since we know L to be regular, then we know there must exists a NFA $M_L = (Q_L, \Sigma_L, \delta_L, q_L, F_L)$ whose language is L. It suffices to construct a NFA, $M_{\overline{L}} = (Q_{\overline{L}}, \Sigma_{\overline{L}}, \delta_{\overline{L}}, q_{\overline{L}}, F_{\overline{L}})$ recognizing $\overline{L}$. We construct such a NFA as follows:

$$\begin{aligned}
&Q_{\overline{L}} = Q_L\\
&\Sigma_{\overline{L}} = \Sigma_L\\
&\delta_{\overline{L}} = \delta_L\\
&q_{\overline{L}} = q_L\\
&F_{\overline{L}} = Q_L - F_L
\end{aligned}$$

Thus, $\overline{L}$ is regular. $\square$

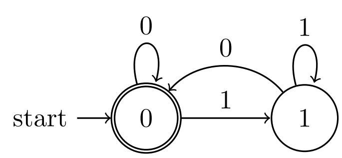
Suppose we have the following NFA:

M_{odd}:



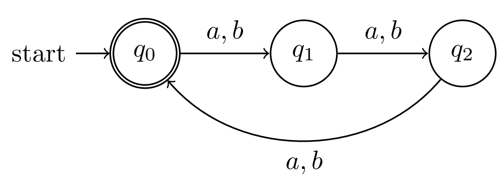
Then its complement is as follows:

$\overline{M_{\text{odd}}}$:



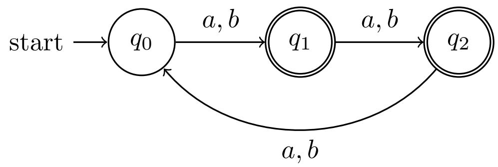
Another example is the following:

M_{3n}:



Then its complement is as follows:

$\overline{M_{3n}}$:



3 Closure under Intersection

So far we have shown that the class of regular languages is closed under union and complement. This is enough to show that the class of regular languages is closed under intersection. We show this by using De Morgan's dualities.

Lemma 4 (Intersection of Regular Languages)**.** *The class of regular languages is closed under intersection.*

Proof. Suppose L_1 and L_2 are regular languages. Then we know by the previous section that $\overline{L_1}$ and $\overline{L_2}$ are also regular, which implies that $\overline{L_1} \cup \overline{L_2}$ is regular. Finally, we know $\overline{\overline{L_1} \cup \overline{L_2}}$ is regular. By De Morgan's dualities we know that $X \cap Y = \overline{\overline{X} \cup \overline{Y}}$, thus, the class of regular languages is closed under intersection. □

4 Closure under concatenation

Suppose L_1 and L_2 are regular languages. Then is the following language also regular?

$$L_\circ = \{w_1 w_2 \mid w_1 \in L_1 \text{ and } w_2 \in L_2\}$$

We call the language $L_\circ$ the concatenation of the languages L_1 and L_2 and is denoted $L_1 \circ L_2$. It turns out that $L_\circ$ is indeed regular.

Lemma 5 (Concatenation of Regular Languages)**.** *The class of regular languages is closed under concatenation.*

Proof. Suppose L_1 and L_2 are regular. It suffices to show that $L_1 \circ L_2$ is regular. We proceed just as we have in the previous results and first suppose $M_1 = (Q_1, \Sigma_1, \delta_1, q_1, F_1)$ and $M_2 = (Q_2, \Sigma_2, \delta_2, q_2, F_2)$ are NFAs that accept L_1 and L_2 respectively. Then we construct an NFA, $M_\circ = (Q_\circ, \Sigma_\circ, \delta_\circ, q_\circ, F_\circ)$, that accepts $L_1 \circ L_2$ as follows:

$$Q_\circ = Q_1 \cup Q_2$$
$$\Sigma_\circ = \Sigma_1 \cup \Sigma_2$$

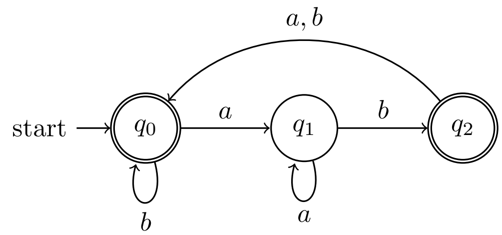
$$\delta_\circ(q, a) = \begin{cases} \delta_1(q, a) & : q \in Q_1, q \notin F_1 \text{ and } a \in \Sigma_1 \\ \delta_1(q, a) & : q \in Q_1, q \notin F_1 \text{ and } a = \epsilon \\ \{q_2\} \cup \delta_1(q, a) & : q \in F_1 \text{ and } a = \epsilon \\ \delta_1(q, a) & : q \in F_1 \text{ and } a \in \Sigma_1 \\ \delta_2(q, a) & : q \in Q_2 \text{ and } a \in \Sigma_2 \\ \delta_2(q, a) & : q \in Q_2 \text{ and } a = \epsilon \end{cases}$$

$$q_\circ = q_1$$
$$F_\circ = F_2$$

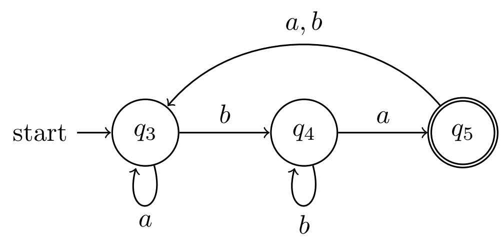
Thus, we obtain our result. □

Consider the following example:

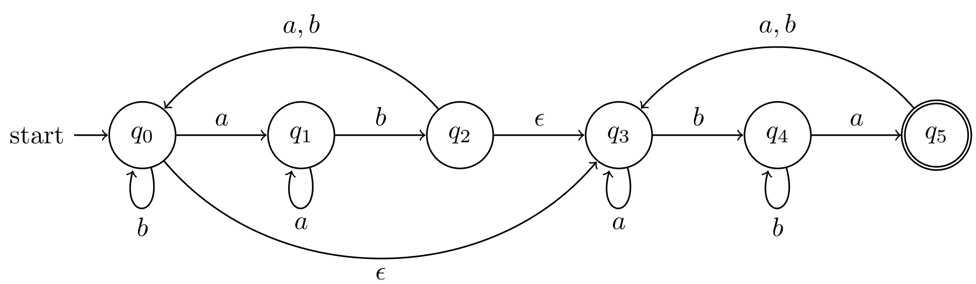
M_{ab}:



M_{ba}:



$M_{ab} \circ M_{ba}$:



5 Closure under Kleene star

We now show that for an regular language, L, the language L^* that consists of every possible concatenation of the words in L is regular. The operation L^* is called the Kleene-star of L.

Lemma 6 (Kleene Star of Regular Languages)**.** *The class of regular languages is closed under the Kleene star.*

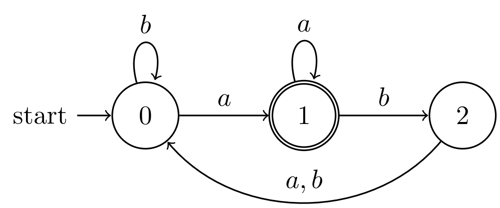
Proof. Suppose L is regular. It suffices to show that L^* is regular. We proceed just as we have in the previous results and first suppose $M = (Q, \Sigma, \delta, q_0, F)$ is a NFA that accepts L. Suppose q_* is a new state such that $q_* \notin Q$. Then we construct an NFA, $M_* = (Q_*, \Sigma_*, \delta_*, q_*, F_*)$, that accepts L^* as follows:

$$
\begin{aligned}
&Q_* = \{q_1\} \cup Q \\
&\Sigma_* = \Sigma \\
&\delta_*(q,a) = \begin{cases} \{q_0\} & : q = q_* \text{ and } a = \epsilon \\ \emptyset & : q = q_* \text{ and } a \neq \epsilon \\ \{q_0\} \cup \delta(a,q) & : q \in F \text{ and } a = \epsilon \\ \delta(q,a) & : q \in F \text{ and } a \neq \epsilon \\ \delta(q,a) & : q \in Q \text{ and } q \notin F \end{cases} \\
&q_* = q_1 \\
&F_* = \{q_1\} \cup F
\end{aligned}
$$

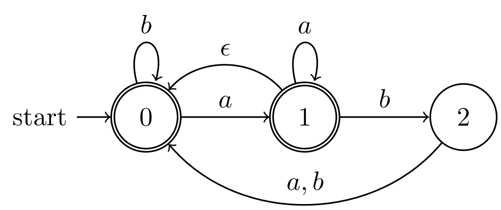
Thus, we obtain our result. □

Note that the state q_* is a new start state such that $q_* \notin Q$. The reader might be wondering why we need to add a new start state. A large number of examples work when we simply make the original start state final, but there are counter examples.

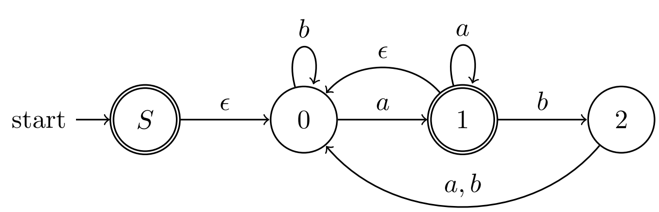
Consider the following NFA:



Computing the Kleene star of the previous example where we do not add a new start state, but make the original final is then:



It is easy to see that the word ab is accepting by the previous NFA, but ab is not in the language of the original NFA, and neither are a and b, thus, it is not the case that ab is a result of the star operation. Therefore, the language of the last NFA is incorrect. However, if we instead add a new start state, then we end up with the following:



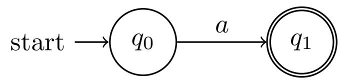
Clearly, the word ab is not accepted.

6 An Example Putting it All Together

Suppose we wanted to prove that the language $L = \{w \mid w \in \{a, b\}^* \text{ and } w = ab^n a \text{ where } n \in \mathbb{N}\}$ is regular. We can prove this by constructing several small NFAs and then using the regular operators on NFAs to construct our final NFA.

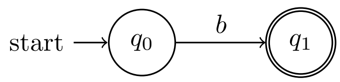
First, we construct an NFA recognizing the language $L_a = \{a\}$:

M_a:



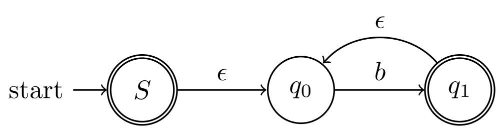
Next we construct the NFA recognizing the language $L_b = \{b\}$:

M_b:



Now we construct the NFA for M_b^* as follows:

M_b^*:



Finally, by concatenating M_a to both sides of M_b^* we can obtain our final NFA:

M_b^*: