Regex to NFA

Noah Buchanan

CS 4043: Formal Languages
University of Arkansas - Fort Smith
Fall 2021

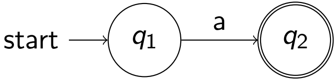
October 24, 2021

We will be converting R into an NFA N. We must consider the six cases in the formal definition of regular expressions.

Say that R is a regular expression if R is

1. a from some a in the alphabet Σ,
2. ϵ,
3. $\emptyset$,
4. $(R_1 \cup R_2)$, where R_1 and R_2 are regular expressions,
5. $(R_1 \circ R_2)$, where R_1 and R_2 are regular expressions, or
6. (R_1^*). where R_1 is a regular expression.

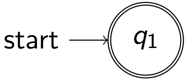
$R = a$ for some $a \in \Sigma$. Then $L(R) = \{a\}$, and the following NFA recognizes $L(R)$.



Note that this machine does not fit the definition of a DFA because it has some states with no exiting arrow for each possible input symbol in the alphabet. An equivalent DFA could have been shown here but an NFA is all we need to describe it in a succint form.

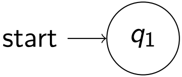
Formally, $N = (\{q_1, q_2\}, \Sigma, \delta, q_1, \{q_2\})$, where we describe δ by saying that $\delta(q_1, a) = \{q_2\}$ and that $\delta(r, b) = \emptyset$ for $r \neq q_1$ or $b \neq a$.

$R = \epsilon$. Then $L(R) = \{\epsilon\}$, and the following NFA recognizes $L(R)$.



Formally, $N = (\{q_1\}, \Sigma, \delta, q_1, \{q_1\})$, where $\delta(r, b) = \emptyset$ for any r and b.

$R = \emptyset$. Then $L(R) = \emptyset$, and the following NFA recognizes $L(R)$.



Formally, $N = (\{q_1\}, \Sigma, \delta, q_1, \emptyset)$, where $\delta(r, b) = \emptyset$ for any r and b.

For the last three cases $R = R_1 \cup R_2$, $R = R_1 \circ R_2$, and $R = R_1^*$. We can use our knowledge of the class of regular languages being closed under the regular operations.

R_1 and R_2 are both regular languages and because of that $R_1 \cup R_2$ is regular. Because this is the case we can create some NFA N to recognize R.

$R_1 \circ R_2$ is also regular because concatenation is a regular operation. Therefore we can create some NFA N to recognize R.

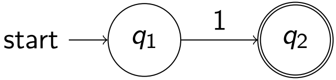
R_1 is a regular language and the star operation is a regular operation. Therefore we can create some NFA N to recognize R.

Now we will be using what we have laid out to formally construct an NFA from the following Regular Expression:

$$(1 \cup 010 \cup 10)^*(00 \cup 11)^+.$$

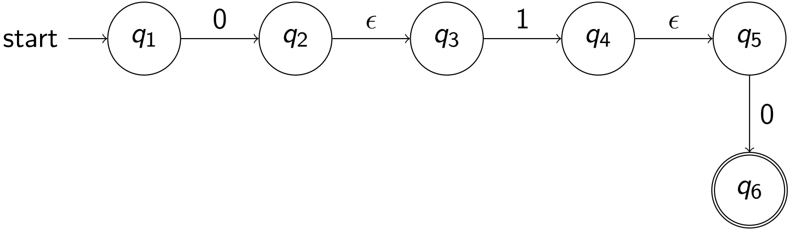
Note: R^+ can be represented as $R \circ R^*$.

We start left to right in the first parentheses with $1 \cup 010$. We cannot do both unions at once because these operations are not unary but binary.



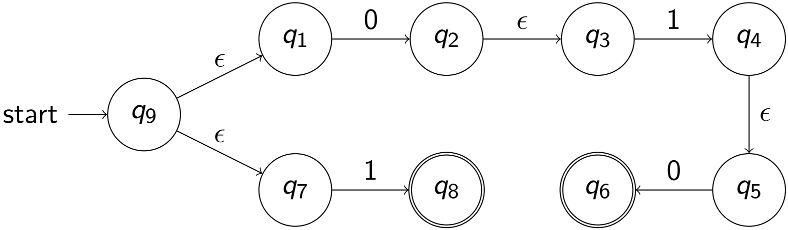
1:

010:

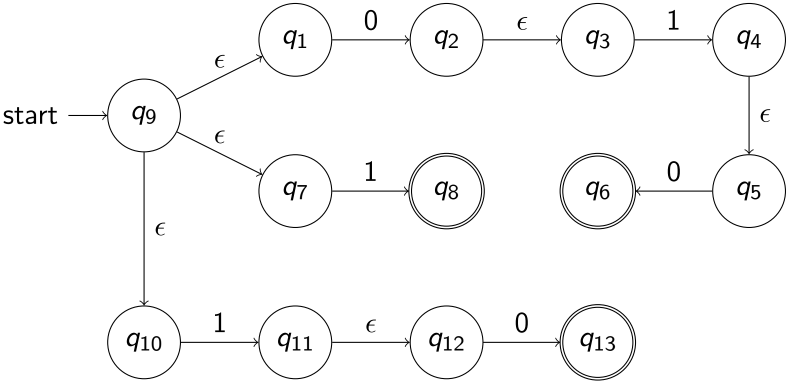


Note that 010 is just the concatenation of simple machines like the first shown.

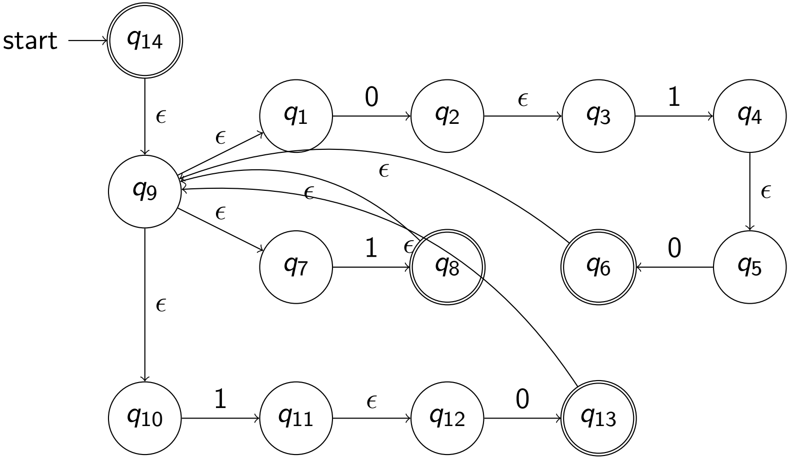
We must now union the two machines:



We then union this machine with 10 on the following slide.

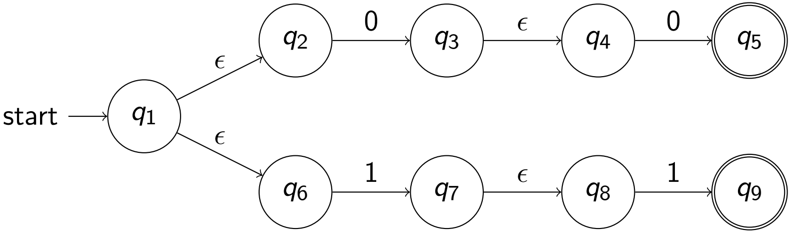

start
q9
q1
q2
q3
q4
q5
q6
q7
q8
q10
q11
q12
q13
0
1
ϵ

To star this NFA is a simple operation. We simply create a new start state that is also an accept state and epsilon transfer all accept states to the previous start state.

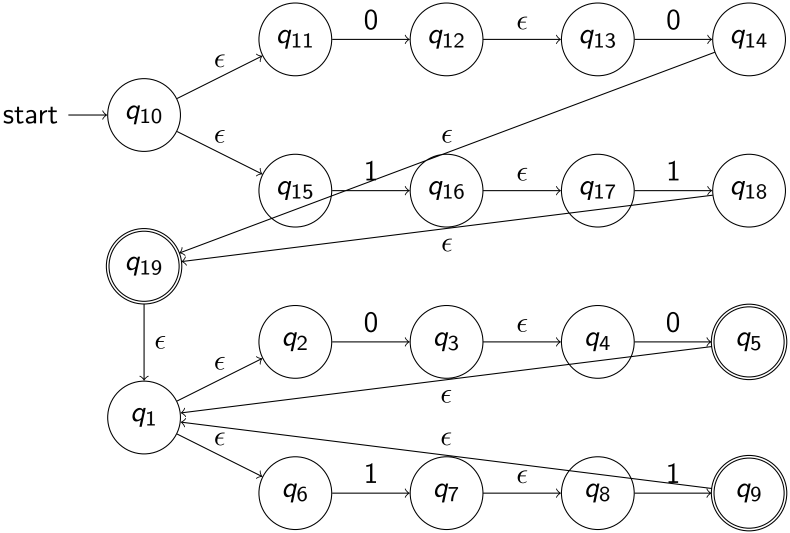


We will be moving to the second parentheses now and concatenate the two machines once both are finished at the end.

As before we create the machines and union them using the same methods as before. I will not detail this step by step at this point.



Remember that $R^+ = R \circ R^*$. We implement this as follows:



Lastly these two machines constructed from the parentheses must be concatenated to formally construct the finished NFA.