return the same set of states as $\delta(q, a)$, the transition function on symbols of the NFA with ϵ-transitions. The latter one includes only the states reachable from q on input symbol a. The former one is more complex - to compute it, we first follow all the ϵ-transitions from q, then all transitions on input symbol a from these states, and finally all the ϵ-transitions from this set of states. To see why these two are different, try calculating $\hat{\delta}(q_0, 0)$ and compare it to $\delta(q_0, 0)$. Note that $\delta(q, a) \subset \hat{\delta}(q, a)$.

8.3 String Acceptance and Language Accepted by a NFA With ϵ-transitions

Now that we have defined properly the transition function for strings for NFAs with ϵ transitions, we can define precisely the string acceptance by a NFA with ϵ-transitions, as well as the language accepted by a NFA with ϵ-transitions. Both definitions are pretty much the same as for the basic NFA.

Definition 8.6: String Acceptance by a NFA With ϵ-transitions

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a NFA with ϵ-transitions. String $w \in \Sigma^*$ is accepted by M if $\hat{\delta}(q_0, w)$ contains at least one accepting state from F.
More formally, w is accepted by M if $\hat{\delta}(q_0, w) \cap F \neq \emptyset$.

Therefore, string w is accepted by a NFA with ϵ-transitions if there exists a sequence of transitions on the successive symbols of w (interspersed with ϵ transitions) from the starting state to one of the accepting states of the automaton.

Definition 8.7: Language Accepted a NFA With ϵ-transitions

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a NFA with ϵ-transitions. Language L accepted by M, denoted by $L(M)$ or L_M, is the set of all strings accepted by M.

$$L(M) = \{w | w \in \Sigma^*, \hat{\delta}(q_0, w) \cap F \neq \emptyset\}$$

8.4 Equivalence Between NFAs and NFAs With ϵ-transitions

As was the case for NFAs and DFAs, adding ϵ-transitions does not increase the power of the automata in terms of the languages they accept. Exactly the same set of languages, the set of regular languages, is also accepted by NFAs with ϵ-transitions.

Theorem 8.1: Equivalence Between NFAs and NFAs With ϵ-transitions

Let N be the set of all languages accepted by NFAs, and E the set of all languages accepted by NFAs with ϵ-transitions. Then $N = E$

In other words, every language accepted by some NFA with ϵ-transitions is also accepted by some NFA and vice versa.

Proof. We again need to prove that $N \subset E$ and $E \subset N$. To prove that $N \subset E$, we need to prove that every language accepted by some NFA is also accepted by some NFA with ϵ-transitions. This is pretty simple, since we can interpret any NFA as a NFA with ϵ-transitions, where ϵ-transitions are missing. If we want to be more formal, we would interpret an NFA with transition function δ as an NFA with ϵ-transitions with transition function δ', such that $\delta'(q, a) = \delta(q, a)$ for every symbol a of the alphabet and every state q of the NFA, and $\delta'(q, \epsilon) = \emptyset$ for every state q of the NFA. Therefore, it is trivial to find an NFA with ϵ-transitions that accepts the same language as a given NFA without ϵ-transitions. Therefore, $N \subset E$.

The other direction is, again, not trivial. Given an NFA with ϵ-transitions M, we need to find an NFA without ϵ-transitions M' that accepts the same language as M. We need to somehow eliminate the ϵ-transitions from M.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be our NFA with ϵ-transitions. We need to design an NFA without ϵ-transitions M' that will somehow simulate M. Again, by "simulate", we mean that M' will be in the same set of states as M after reading the same string (provided, obviously, that they start reading the string in the same state). The idea here is to keep the states (including the start state) the same, but change the transition function so that it follows the ϵ-transitions. So, our NFA without ϵ-transitions will be $M' = (Q, \Sigma, \delta', q_0, F')$. We will see later on why we also need to change the set of accepting states, so we cannot just take the same set F as for M. But let us focus first on the transition function δ' of our NFA without ϵ-transitions. The idea is for $\delta'(q, a)$, where $a \in \Sigma$, to be the set of states in which M will be if it is in the state q and reads a single input symbol a. What states will M be in after it reads a single input symbol a? To find this set of states, we first need to follow all ϵ-transitions from q, then from all these states follow the transitions on input symbol a, and then from all these resulting states follow all ϵ-transitions. Luckily, we already have something that denotes the set of all states reachable from some state q using only ϵ-transitions - the ϵ-CLOSURE(q)! So, then we have that

$$\delta'(q, a) = \epsilon\text{-CLOSURE}(\bigcup_{i=1}^{n} \delta(p_i, a)),$$

where $\{p_1, p_2, \ldots, p_m\}$ is ϵ-CLOSURE$(()q)$. But this is exactly what $\hat{\delta}(q, a)$ is. As a reminder, $\hat{\delta}(q, a)$ is the transition function for strings for M, applied to a string that consists only of a single symbol a. So, let us define

$$\delta'(q, a) = \hat{\delta}(q, a).$$

This is exactly what we need - the set of states in which the NFA with ϵ-transitions M will be if it reads the string comprising only symbol a.

So, it looks like the NFA without ϵ-transitions M' defined in this way simulates the NFA with ϵ-transitions M. But, again, to be absolutely sure, we would need to prove that M' is in exactly the same set of states as M when they both start in the same state and read the same string. In other words, we would need to prove that

$$\hat{\delta'}(q, w) = \hat{\delta}(q, w), \text{ for every state } q \in Q \text{ and } \textit{almost} \text{ every string } w \in \Sigma^*.$$

On the left we have a transition function for strings for our NFA without ϵ-transitions, and on the right a transition function for strings for our NFA with ϵ-transitions. The "almost" part above refers to the empty string - the equation does not hold for the empty string. For $w = \epsilon$, we have that $\hat{\delta'}(q, \epsilon) = \{q\}$, while $\hat{\delta}(q, \epsilon) = \epsilon$-CLOSURE$(q)$. These two are not the same if there exist epsilon transitions out of q.

Proving this is surprisingly fiddly and non-trivial, and as such is left for Appendix **??** for interested readers. We will just proceed as if we have proven this, and hence proven that M' indeed simulates M.

The only thing left to complete our NFA without ϵ-transitions M' is to determine its set of accepting states F'. Intuitively, since its transition function δ' simulates precisely the transition function of M, we might think that it surely has to be $F' = F$, so that we can just take the same set of accepting states as in M. This is almost true, but not exactly. The problem is that we have defined δ' for letter of the alphabet Σ only, but not for ϵ too (because there are no ϵ-transitions in M'). So, in what states would M' then be if it reads ϵ, starting from its start state q_0? This would be $\hat{\delta'}(q_0, \epsilon)$, which is, by definition of $\hat{\delta'}$, just $\{q_0\}$. On the other hand, the states in which M is after it reads ϵ in state q_0 is ϵ-CLOSURE(q_0). It is, therefore, possible for M to accept the empty string (if ϵ-CLOSURE(q_0) contains one of the accepting states from F), and for M' to reject it (if q_0 is not an accepting state). In this case, M and M' would not accept the same language. Therefore, we need to define the set of accepting state F' of M' in the following way:

$$F' = \begin{cases} F, & \text{if } \epsilon\text{-CLOSURE}(q_0) \text{ does not contain one of the accepting states} \\ F \cup \{q_0\}, & \text{if } \epsilon\text{-CLOSURE}(q_0) \text{ contains one of the accepting states} \end{cases}.$$

Now we have accounted for the case of the ϵ, and we know that for any other string w and any state q, $\hat{\delta'}(q, w) = \hat{\delta}(q, w)$, therefore it is very easy to prove that M and M' accept exactly the same language. And with this, we have constructed the NFA M' without ϵ-transitions that accepts exactly the same language as M. And, hence, we have proven that any language accepted by some NFA with ϵ-transitions is also accepted by some NFA without ϵ-transitions, which proves that $E \subset N$, and completes the proof of our theorem. □

We will summarise once again our construction. Let L be the language accepted by a NFA with ϵ-transitions $M = (Q, \Sigma, \delta, q_0, F)$. Then NFA M' without ϵ-transitions, defined as $M' = (Q, \Sigma, \delta', q_0, F')$, where

$$\delta'(q, a) = \hat{\delta}(q, a) \text{ for every state } q \in Q \text{ and every symbol } a \in \Sigma,$$

and

$$F' = \begin{cases} F, & \text{if } \epsilon\text{-CLOSURE}(q_0) \text{ does not contain one of the accepting states} \\ F \cup \{q_0\}, & \text{if } \epsilon\text{-CLOSURE}(q_0) \text{ contains one of the accepting states} \end{cases},$$

also accepts the language L.

A particularly nice thing about this construction is that the resulting NFA without ϵ-transitions has exactly as many states as the original NFA with ϵ-transitions, just more transitions.

Video 8.3: NFA With ϵ-transitions to NFA Without ϵ-transitions

Using the construction from Theorem 8.1, find the NFA M' that accepts the same language as NFA with ϵ-transitions M' from the Example 8.2