Procedure NFA to DFA

Note: The following page is another method for converting an NFA to a DFA. I will first present the procedure and then show an example from the outline procedure.

Let's first start with the actual outline of the procedure[1].

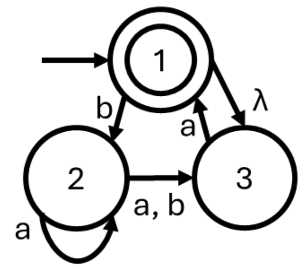
Procedure for Converting an NFA to its equivalent DFA

1. Create a graph G_D with vertex $\{q_0\}$. Make this vertex the starting state of your FSA.
2. Repeat the following step until NO more edges are missing.
   Take any vertex $\{q_i, q_j, ..., q_k\}$ of G_D that has no outgoing edge for $a \in \Sigma$. Compute $\delta_N^*(q_i, a), \delta_N^*(q_j, a), ..., \delta_N^*(q_k, a)$. If
   $$\delta_N^*(q_i, a) \cup \delta_N^*(q_j, a) \cup ... \cup \delta_N^*(q_k, a) = \{q_l, q_m, ..., q_n\},$$
   create a vertex G_D labeled $\{q_l, q_m, ..., q_n\}$ if it does not already exist.
   Add to G_D an edge from $\{q_i, q_j, ..., q_k\}$ to $\{q_l, q_m, ..., q_n\}$ and label it with a.
3. Every state G_D whose label contains any $q_f \in F_N$ is identified as a final vertex.
4. If M_N accepts λ or ϵ, the vertex $\{q_0\}$ in G_D is also made a final vertex.

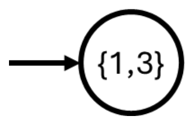
[1] Procedure comes from the book *An Introduction to Formal Languages and Automata* by Peter Linz Sixth Edition.

Example from Slide Notes on Nondeterminism (page 24)

Convert the following NFA into its equivalent DFA using the outline procedure from the previous page.



1. Create the starting state for G_D. We see here that we initially start at state q_1, however we have a λ transition from q_1 to q_3. That means our starting will be labeled q_{13}

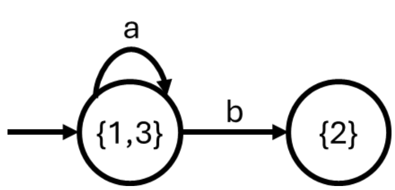


2. Now we go through the vertices in G_D and compute the δ^* for each symbol $a \in \Sigma$. Here, $\Sigma = \{a, b\}$. Initially we have two states already incorporated into the potential DFA. We have state q_1 and q_3. That means we will compute $\delta^*(\{q_1, q_3\}, a)$ and $\delta^*(\{q_1, q_3\}, b)$.

   $\delta^*(\{q_1, q_3\}, a) = \{q_1, q_3\}$

   $\delta^*(\{q_1, q_3\}, b) = \{q_2\}$

   From our 1st iteration, we have 1 new state $\{q_2\}$ to add to G_D.

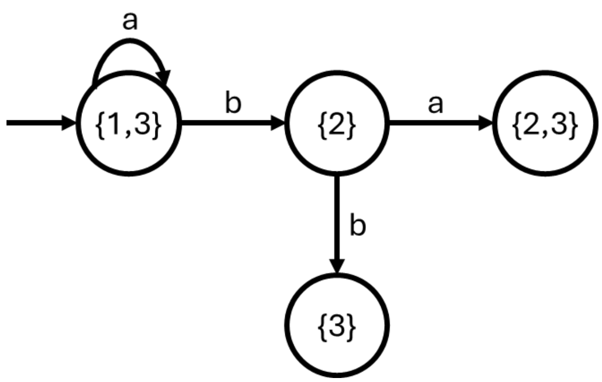


   Since we have one new state, we need to repeat this step by computing δ^* for $\{q_2\}$.

   $\delta^*(q_2, a) = \{q_2, q_3\}$

   $\delta^*(q_2, b) = \{q_3\}$

From our 2nd iteration, we have 2 new states $\{q_3\}$ and $\{q_2, q_3\}$ to add to G_D.



Since we have two new states, we need to repeat this step by computing δ^* for $\{q_3\}$ and $\{q_2, q_3\}$.

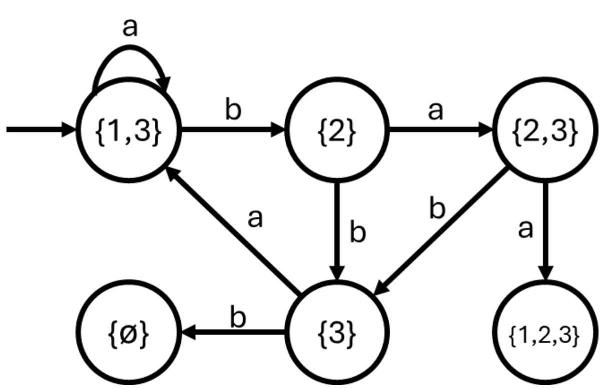
$\delta^*(q_3, a) = \{q_1, q_3\}$

$\delta^*(q_3, b) = \{\emptyset\}$ This represents our trap state!

$\delta^*(\{q_2, q_3\}, a) = \{q_1, q_2, q_3\}$

$\delta^*(\{q_2, q_3\}, b) = \{q_3\}$

From our 3rd iteration, we have 2 new states $\{q_1, q_2, q_3\}$ and $\{\emptyset\}$ to add to G_D.



Since we have two new states, we need to repeat this step by computing δ^* for $\{\emptyset\}$ and $\{q_1, q_2, q_3\}$.

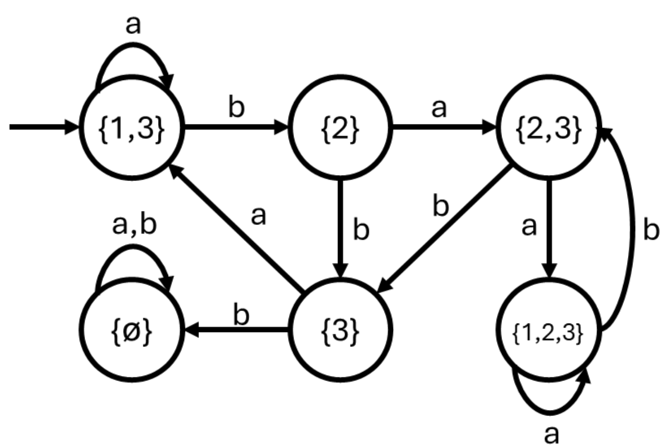
$\delta^*(\{\emptyset\}, a) = \{\emptyset\}$ This represents our trap state!

$\delta^*(\{\emptyset\}, b) = \{\emptyset\}$ This represents our trap state!

$\delta^*(\{q_1, q_2, q_3\}, a) = \{q_1, q_2, q_3\}$

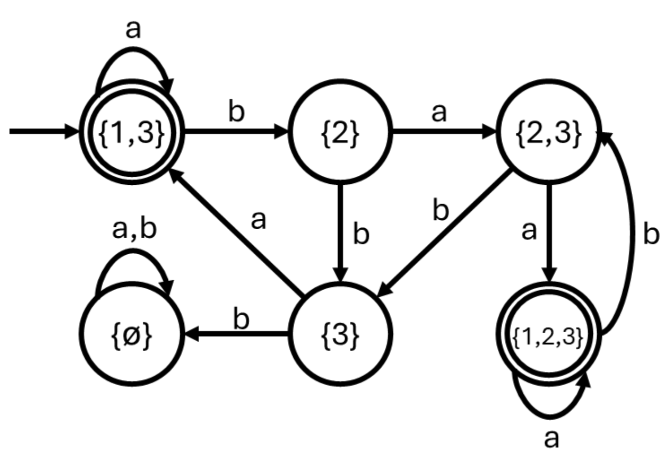
$\delta^*(\{q_1, q_2, q_3\}, b) = \{q_2, q_3\}$

From our 4th iteration, we have no new states to add to G_D.

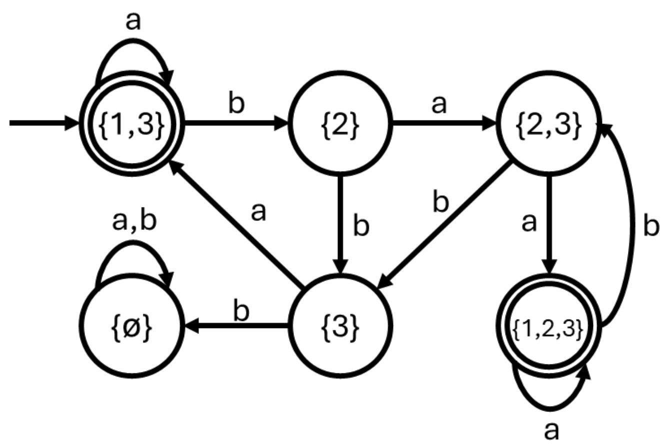


Since there are no more new states, we are done with step 2 since we have computed all possible transition functions that reaches a state.

3. Now we have to add the final states to G_D from the NFA. The example has only 1 final state q_1. That means any state that is associated with q_1 is considered a final state. The DFA we have computed has two states that are associated with q_1. Those states are $\{q_1, q_3\}$ and $\{q_1, q_2, q_3\}$.



4. The final step is to check to see if empty string (λ and ϵ) is accepted in the NFA. If it is, then we make the starting state a final state. Now in this particular example, the starting state is already the final state due to step 3, but it is interesting to point out that λ or ϵ is also accepted as well. However we do not need to make any updates to our DFA. This now results in the final DFA that is equivalent to the NFA.



Notice that our final DFA only has 6 states, but the powerset of the original NFA contains 8 states ($2^3 = 8$). That means there are 2 states that are unreachable. Those states are $\{q_1\}$ and $\{q_1, q_2\}$.