Examples of NFA Constructions

These exercises are taken from *Introduction to the Theory of Computation* by Michael Sipser.

Example 1

Let A be a regular language over an alphabet Σ. Define DROP-OUT(A) to be the language consisting of all strings that can be obtained by removing one symbol from a sting in A. Explicitly,

$$\text{DROP-OUT}(A) = \{xz \mid xyz \in A, x, z \in \Sigma^*, y \in \Sigma\}.$$

Show that DROP-OUT(A) is regular.

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA recognizing A, we create an NFA M' that recognizes DROP-OUT(A). Pictorially, we have two copies of M and we add an ε-transition from every state in the first copy to the states in the second copy that are normally attainable. More formally, we have $M' = (Q \times \{0,1\}, \Sigma, \delta', (q_0, 0), F \times \{1\})$ with δ' defined as follows.

For all $q \in Q$, $s \in \Sigma$,

$$\delta'((q,0),s) = \{(\delta(q,s),0)\} \cup \{(r,1) \mid \exists s' \in \Sigma, \delta(\delta(q,s),s') = r\}$$
$$\delta'((q,1),s) = \{(\delta(q,s),1)\}.$$

Briefly, this machine simulates the DFA on the input and non-deterministically chooses to read a symbol that is not there at some point in the computation. When it does that, it moves to the second copy of the machine where it has no ε-transitions, so it must read the rest of the input normally. Note that the non-determinism is really important here as the machine can simulate any missing letter it wants at any time (although only once during the computation).

We now have to formally prove that this M' recognizes DROP-OUT(A).

If $w \in$ DROP-OUT(A), we can write it as xz where $x, z \in \Sigma^*$ and there exists $y \in \Sigma$ such that $xyz \in A$. Therefore, there is a branch of computation of M' that reads the whole string x without taking any ε-transition and then takes the ε-transition corresponding to reading y before reading the whole string z in the second copy. After all this computation, M' is in the same state as M after reading xyz so it accepts w.

If M' accepts w, then M' is in the second copy of the DFA, so it must have taken an ε-transition, say it corresponds to reading the symbol $y \in \Sigma$. We can look at the branch of computation that accepts to write w as xz where x was read in the first copy and z was read in the second copy. We conclude that M accepts xyz, so $w \in$ DROP-OUT(A).

This finishes the proof that DROP-OUT(A) is regular as it is recognized by an NFA.

Example 2[1]

Let B and C be regular languages over $\Sigma = \{0, 1\}$. Define

$$B \overset{1}{\leftarrow} C = \{w \in B \mid \exists y \in C, w \text{ and } y \text{ contain equal numbers of } 1\text{s}\}.$$

Show that this language is regular.

We also define

$$\text{ONE}(C) = \{w \in \Sigma^* \mid \exists y \in C,\ w \text{ and } y \text{ have the same number of } 1\text{s}\}.$$

If we show that $\text{ONE}(C)$ is regular, then it will imply $B \overset{1}{\leftarrow} C$ is regular because $B \overset{1}{\leftarrow} C = \text{ONE}(C) \cap B$ and the class of regular languages is closed under intersection[2].

Let $M = (Q, \Sigma, \delta, q_0, F)$ be a DFA that recognizes C, we will construct an NFA M' that recognizes $\text{ONE}(C)$. The idea is that M' will simulate M without taking into account the 0's of the input and bypassing the 0 transitions in M. This will recognize $\text{ONE}(C)$ because we only care about the 1s in this language. Formally, we have $M' = (Q, \Sigma, \delta', q_0, F)$, where δ' is defined as follows[3]:

For all $q \in Q$,

$$\delta'(q, 1) = \{\delta(q, 1)\} \cup \{r \in Q \mid \exists k \in \mathbb{N}, \delta^*(\delta(q, 1), 0^k)\}$$
$$\delta'(q, 0) = \{q\}$$

If $w \in \text{ONE}(C)$, then there exists $y \in C$ with the same number of 1's and on input w, there is a branch of computation of M' that does the exact same changes of states as M on y (except maybe some transition from a state to itself when reading a 0), hence this branch accepts w.

If M' accepts w, we will transform w to a word w' in C without adding any 1's. Consider the accepting branch of computation, for each transition arrow taken that was not in M, do the following: if the arrow is a ε-transition, add a 0 to w in this place, otherwise, if it was a 0-transition, remove the 0 that was consumed for the transition. After these transformations, w' will be accepted by M, so $w \in \text{ONE}(C)$.

This finishes the proof that $\text{ONE}(C)$ is regular.

[1] The book contains an alternate solution.

[2] This will probably be proven in class, otherwise, a proof can easily be found online.

[3] δ^* denotes the the transition function extended to Σ^* in the usual way, i.e.: $\delta^*(q, aw) = \delta^*(\delta(q, a), w)$ for any $q \in Q$, $w \in \Sigma^*$ and $a \in \Sigma$.