CS 228 (M) - Logic in CS
Tutorial VI - Solutions

Ashwin Abraham

IIT Bombay

12th September, 2023

Table of Contents

ε-NFAs

An ε-NFA is a generalization of an NFA wherein you are allowed to make some transitions without reading any alphabets in the word. These transitions are known as ε-transitions. To be precise, an ε-NFA is a tuple $(Q, \Sigma, \Delta, I, F)$ where Q is a finite set of states, $I \subseteq Q$ is the set of initial states, $F \subseteq Q$ is the set of final states and $\Delta \subseteq Q \times \Sigma \cup \{\varepsilon\} \times Q$ is the transition relation. The intuition here is that we are now allowed to read the empty word ε and possibly make a transition, and since ε can be inserted anywhere in a word without changing it, we can make some transitions without reading any alphabets.

To formalize this, we define the ε-closure $C_\varepsilon(q)$ of a state q as the set of states reachable from q via only ε-transitions. We define the ε-closure of a set of states as the union of the ε-closures of its members. We define $\hat{\Delta} : 2^Q \times \Sigma^* \to 2^Q$ recursively as follows:

(1) $\hat{\Delta}(S, \varepsilon) = C_\varepsilon(S)$ for any $S \subseteq Q$

(2) $\hat{\Delta}(S, xa) = C_\varepsilon(\{q' : \exists q \; q \in \hat{\Delta}(S, x) \wedge (q, a, q') \in \Delta\})$ for any $S \subseteq Q$, $x \in \Sigma^*$ and $a \in \Sigma$

A word $w \in \Sigma^*$ is accepted by the ε-NFA iff $\hat{\Delta}(I, w) \cap F \neq \emptyset$

Equivalence of ε-NFAs and DFAs

The set of languages accepted by ε-NFAs is precisely the set of regular languages. Firstly, note that since every NFA is also an ε-NFA, this set must be a superset of the set of regular languages. For every ε-NFA, we show that we can construct a DFA accepting the same language - this shows that the set of languages accepted by ε-NFAs is precisely the set of regular languages.
Given an ε-NFA $(Q, \Sigma, \Delta, I, F)$, construct the DFA $(Q', \Sigma, \delta, C_\varepsilon(I), F')$ where $Q' = \{C_\varepsilon(S) : S \subseteq Q\}$ and $\delta : Q' \times \Sigma \to Q'$ is such that $\delta(S, a) = C_\varepsilon(\{q' : \exists q \; q \in S \wedge (q, a, q') \in \Delta\})$. It can easily be shown by induction that $\hat{\delta}(C_\varepsilon(I), w) = \hat{\Delta}(I, w)$ for any $w \in \Sigma^*$. We define $F' = \{S \in Q' : S \cap F \neq \emptyset\}$. We can now see that $\hat{\delta}(C_\varepsilon(I), w) \in F'$ iff $\hat{\Delta}(I, w) \cap F \neq \emptyset$. Therefore, it is clear that the constructed DFA accepts the same language as the ε-NFA. $\square$

Question 1

(a) Given $A = (Q, \Sigma, \delta, q_0, F)$, construct the NFA $B = (Q, \Sigma, \Delta, F, \{q_0\})$, where $\Delta = \{(p, a, q) \in Q \times \Sigma \times Q : \delta(q, a) = p\}$.
This NFA is formed by reversing all the transitions of our original DFA and swapping its initial and final states. It is easy to see that a string is accepted by B if and only if its reverse is accepted by A. Therefore the language accepted by B is the set of reverses of the words in $L(A)$, ie $L(B) = L(A)^R = \{w^R : w \in L(A)\}$. Therefore, $L(A)^R$ is regular.

(b) Given a language $L \subseteq \Sigma$ and a set $\Sigma' \subseteq \Sigma$, we can define $L \downarrow \Sigma' \subseteq \Sigma'^*$ as the set of words formed by removing the alphabets not in Σ' from the words in L. If L is regular, ie if it is accepted by an NFA A, then construct the ε-NFA A' by replacing all the transitions in A corresponding to alphabets not in Σ' by ε-transitions. It is easy to show that A' accepts the language $L \downarrow \Sigma'$, which must therefore be regular.

Question 1

(c) The idea over here is to add an extra state that will serve as the sole accepting state. For every transition from a state to a terminal state in the old automata we add a transition from that state to the new state. The new state we add has no outgoing transitions. Formally, if our automaton is $(Q, \Sigma, \Delta, I, F)$, our new automaton will be $(Q \cup \{f\}, \Sigma, \Delta', I', \{f\})$ where $I' = I \cup \{f\}$ if $I \cap F \neq \emptyset$ and I otherwise, and $\Delta'(s, a) = \Delta(s, a) \cup \{f\}$ if $\Delta(s, a) \cap F \neq \emptyset$ and $\Delta(s, a)$ otherwise. Note that $\Delta(f, a) = \emptyset$. Now, for any word w, if w is accepted, then $\hat{\Delta}'(I, w) = \hat{\Delta}(I, w) \cup \{f\}$, and if not, $\hat{\Delta}'(I, w) = \hat{\Delta}(I, w)$. This can be proven by induction on the length of w, and it clearly shows that the languages of the old and new NFAs are the same. □

A similar construction can be used to convert the NFA into an equivalent one with a single initial state. Note that a DFA cannot in general be converted into an equivalent DFA with a single initial state (Try to prove this!)

Question 1

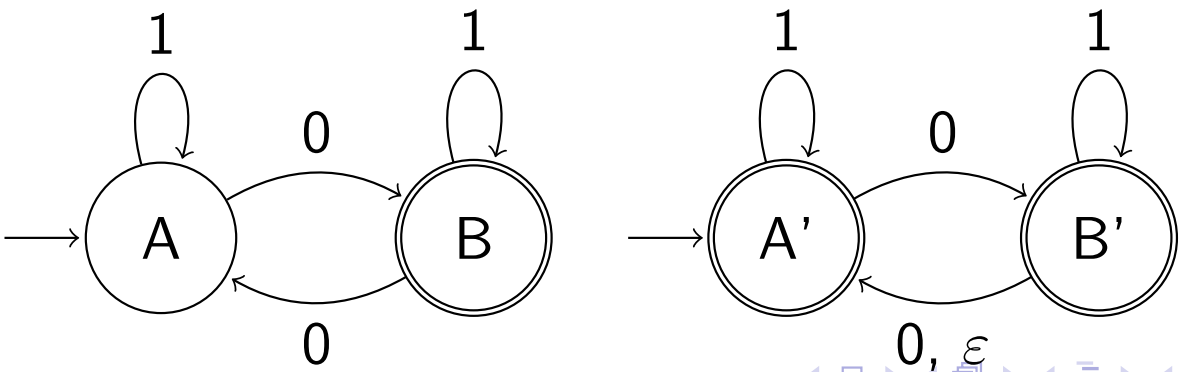
(d) No. Firstly, note that $L(N)^* \subseteq L(N_1)$. This is because:

(1) $\varepsilon \in L(N_1)$ (because $q_0 \in F_1$)

(2) $L(N) \subseteq L(N_1)$ (a run accepted in N will also be accepted in N_1)

(3) $L(N_1)$ is closed under concatenation - to see this, note that if w is accepted by N_1, then there will always be a run of w ending in q_0, due to the ε-transitions from the other final states to q_0. Therefore, if w_1 and w_2 are both accepted by N_1, then so is $w_1 w_2$ due to the run $q_0 \xrightarrow{w_1} q_0 \xrightarrow{w_2} q_0$.

However, if there is a path in N from q_0 to itself, and q_0 is not already final in N, then adding q_0 to F may introduce words in the $L(N_1)$ that may not be present in $L(N)^*$. For example if N is the DFA on the left, then N_1 will be the NFA on the right



Question 1

Note that $1^n \notin L(N)^*$ but $1^n \in L(N_1)$, for any $n > 0$. However, if q_0 is already final in N or if there is no path in N from q_0 to itself, then $L(N_1)$ is indeed $L(N)^*$.

(e) The idea here is to traverse the DFA of L forwards and backwards at the same time. To formalize this, if $(Q, \Sigma, \delta, q_0, F)$ is a DFA accepting L, then consider the DFA $(Q \times 2^Q, \Sigma, \delta', (q_0, F), F')$ where $\delta'((q, S), a) = (\delta(q, a), \{p \in Q : \exists b \in \Sigma\ \delta(p, b) \in S\})$ and $F' = \{(q, S) \in Q \times 2^Q : q \in S\}$. For this DFA, it can be shown that $\hat{\delta}'((q_0, F), x) = (\hat{\delta}(q_0, x), \{p \in Q : \exists y\ |y| = |x| \wedge \hat{\delta}(p, y) \in F\})$ (induct on the length of x). Now, $\hat{\delta}'((q_0, F), x) \in F'$ iff $\exists y\ |y| = |x| \wedge \hat{\delta}(\hat{\delta}(q_0, x), y) = \hat{\delta}(q_0, xy) \in F$, ie $x \in L_{\frac{1}{2}}$. □

Question 2

We construct a DFA whose states are $\{0, 1, \ldots n-1\}$, where being in state i after reading string x means that $\mathit{binary}(x) \equiv i \mod n$. Note that $\mathit{binary}(\varepsilon)$ is taken to be 0 and hence the initial state of this DFA is 0. Now, if $\hat{\delta}(0, x) = i$, then clearly $\hat{\delta}(0, x0) = (2i)\%n$ and $\hat{\delta}(0, x1) = (2i+1)\%n$. Therefore, our transition function δ is such that $\delta(i, 0) = (2i)\%n$ and $\delta(i, 1) = (2i+1)\%n$. Now, if we are to accept strings corresponding to binary numbers that are multiples of n, then clearly our final state will be 0. The DFA we constructed clearly accepts L_n and hence L_n is regular. □

Question 3

We show that every devilish NFA can be reduced to an equivalent DFA by a modified subset construction. Consider the NFA $(Q, \Sigma, \Delta, I, F)$ where a word x is accepted if and only if all runs[1] of x end up in a good state, ie $\hat{\Delta}(I, x) \subseteq F$. Our constructed DFA is $(2^Q, \Sigma, \delta, I, 2^F)$ where $\delta(S, a) = \bigcup_{s \in S} \Delta(s, a)$. It can be shown by induction that $\hat{\delta}(I, x) = \hat{\Delta}(I, x)$, and a word x is accepted by our DFA if and only if $\hat{\delta}(I, x) \in 2^F$, ie $\hat{\Delta}(I, x) \subseteq F$, ie if and only if x is accepted by the devilish NFA. Since our devilish NFA has a DFA accepting the same language as it, the language it accepts must be regular. □

[1] We take this to exclude runs that terminated early on due to lack of transitions - if these are to be included, then we first modify the NFA to include a trap state, and then perform the same construction.

Question 4

Let $(Q, \Sigma, \delta, q_0, F)$ be the DFA accepting L. Construct the NFA $(Q, \Sigma, \Delta, \{q_0\}, F)$ where $\Delta(s, a) = \{\delta(s, a)\}$, for $s \notin F$ and $\Delta(s, a) = \emptyset$ for $s \in F$ (the outgoing transitions from all the final states have been removed). Clearly, if a word $w \in L$ has no proper prefix in L, then it is accepted by the NFA, but if it does (or if the word is not in L), then it isn't accepted. The language accepted by this NFA is therefore nothing but L', and hence L' is regular. $\square$

Question 5

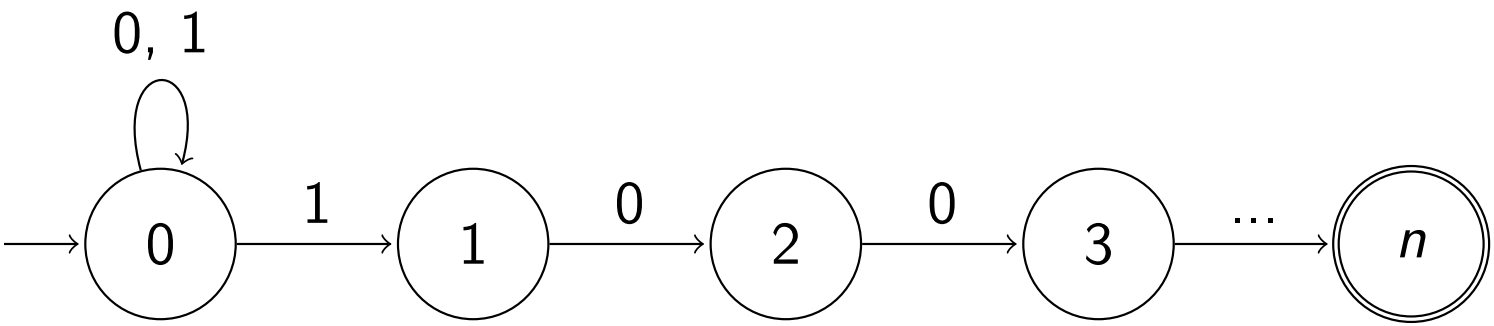
The intuition behind the DFA having at least 2^n states is that it has to "remember" the last n digits, and to do this it must have at least 2^n states. To prove this formally, consider the set of binary strings of length n: $\mathcal{S} = \{\ 00\ldots00,\ 00\ldots01,\ \ldots,\ 11\ldots11\ \}$. Now, for any DFA accepting L_n and any distinct $x, y \in \mathcal{S}$, $\hat{\delta}(q_0, x) \neq \hat{\delta}(q_0, y)$, ie the automaton cannot come to the same state after reading x and y.

Proof.

Since $x \neq y$, without loss of generality, assume that they differ at the i^{th} bit from the right, where x has a 1 and y has a 0. For a contradiction, assume that $\hat{\delta}(q_0, x) = \hat{\delta(q_0}, y)$. Consider the strings $x' = x0^{n-i}$ and $y' = y0^{n-i}$. x' has n^{th} bit from the right as 1 whereas y' has it as 0, hence x' must be accepted and y' cannot be accepted. However, since $\hat{\delta}(q_0, x') = \hat{\delta}(\hat{\delta}(q_0, x), 0^{n-i})$ and $\hat{\delta}(q_0, y') = \hat{\delta}(\hat{\delta}(q_0, y), 0^{n-i})$ and $\hat{\delta}(q_0, x) = \hat{\delta}(q_0, y)$, we will have $\hat{\delta}(q_0, x') = \hat{\delta}(q_0, y')$, ie x' is accepted if and only if y' is accepted which is a contradiction. □

Question 5

Since there are 2^n distinct elements of $\mathcal{S}$, which must all end up in distinct states, any DFA accepting $\mathcal{S}$ must have at least 2^n states.
Note that this language is indeed regular, as it is accepted by the following NFA with $n + 1$ states:



This is an example of an NFA for which any equivalent DFA has exponentially more states.