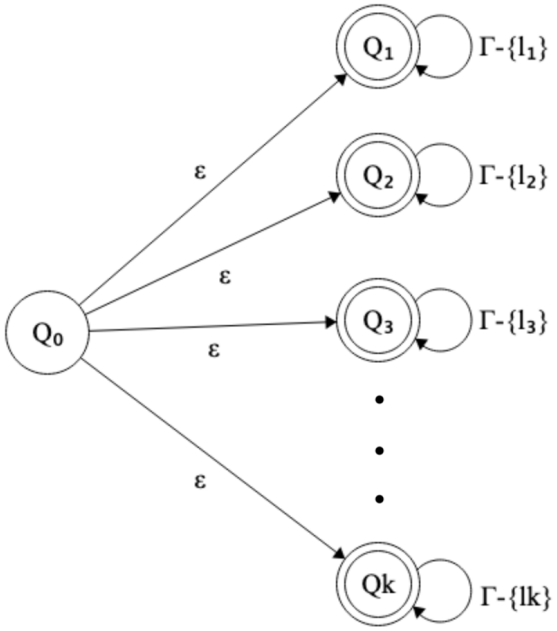
Introduction to Automata and Theory of Computation
COL352 - Assignment 1

Tanishq Dubey (2019CS51077), Satyam Kumar Modi
(2019CS50448), Rupanshu Shah (2019CS10395)

January 2022

Contents

1 Excluding Characters



Required NFA with $(k+1)$ states

Proof for validity of NFA Let the language be L_1 of the grammar and L_2 be the language accepted by the NFA.

- **Part 1:** We first show that $L_1 \subset L_2$. Consider a string $x \in L_1$. We show that $x \in L_2$. For this, we know x does not contain all the letters of the alphabet Γ. Let l_i be one of the letter of alphabet not in x. Let the automaton M take x to state q_i as the first ϵ transition(which is a valid transition). Now, since x does not contain l_i, it will remain in the state q_i until the end of the string. Thus, q_i being a final state means the automaton M accepts string x. Thus, this implies $L_1 \subset L_2$.
- **Part 2:** We now show that $L_2 \subset L_1$. Consider a string $x \in L_2$. We show that $x \in L_1$. For this, we know that x is accepted by the automaton M. Let us consider x contains all the letters from the alphabet Γ. Let us assume that the first transition occurs to state q_i for some i. Now, for M to accept x , it must not have letter l_i in x otherwise it won't be accepted by the automaton. Thus, we conclude that letter l_i is not present in x. This is a contradiction

to the assumption that x contain all letters from Γ. This implies $x \in L_1$. Thus, we have $L_1 \subset L_2$.

This proves $L_1 = L_2$.

No, there do not exist any NFA which accepts L_1. The proof is as follows. Assume there exist an NFA M which accepts L_1. Consider k strings $\{s_1, s_2, ..., s_k\}$ where s_i is of the form $l_1..l_{i-1}l_{i+1}..l_k$. Note s_i does not contain letter l_i. Thus, each s_i belongs to L_1 which imply they belong to $L(M)$. We hvae following two cases possible:

Case I: These k strings land up on k-distinct final states. Since the NFA has k states and all k strings land up in distinct states implies all the states are final states. Let s_i be the string that land up in the initial state(which is also a final state). Since NFA accepts string l_i, the NFA must accept the string $s_i l_i$ as after running through s_i, we land up back to the initial state and from the initial state, l_i takes us to one of the final states(as l_i is accepted by M). But, $s_i l_i$ can't lie in L_1 as it contains all the letters from Γ. Thus, we arrive at a contradiction. So, this case is not possible.

Case II: Some two of these strings land up to the same final state(say F). Let those strings be s_i and s_j for some i and j $(i \neq j)$. Now, consider the string $x = s_j l_i$ and $y = s_i l_i$. Since, x does not have letter l_j, it must be accepted by the NFA M. Note that since s_i and s_j land up to the same final state F, we have $s_j l_i$ and $s_i l_i$ also landing to the same final state(say $F^{'}$). But this means $y = s_i l_i$ is accepted by the NFA M. But this is not possible as y contains all the letters from the alphabet Γ. So, this case is also not possible.

Therefore, we arrive at a contradiction that no such $k - state$ NFA is possible.

2 NFA and Regular Languages

all-NFA $M = (Q, \Sigma, \delta, q_0, F)$. Let $L_1 = L(M)$ be the language defined by all-NFA M. We wish to prove that L_1 is a regular language. Let L_1^C be the complement of Language L_1. If we can prove that L_1^C is a Regular Language, then by closure under complement L_1 is a Regular Language as well.

Let w be a word accepted by M. This implies, on scanning w by M all possible paths end in an accepting state $\in F$

Let us define an NFA $A = (Q, \Sigma, \delta, q_0, Q - F)$. i.e., accept states of M are non-accepting states of A and vice-versa. Let w be any word $\in L_1$. On scanning w by A, all possible paths end in some state $q_f \in F$. Since $q_f \notin Q - F$, therefore, on scanning word w by NFA A, it is rejected.

Now, let w' be any arbitrary word $\in \Sigma^* - L_1$. w' is not accepted by M. This implies on scanning w' by M "at least one" path ends in some state $q_1 \notin F$. This implies $q_1 \in Q - F$. Thus, on scanning word w' by NFA A, there is "at least one" path that ends in an accepting state $q_1 \in Q - F$ of NFA A. Therefore, w' is accepted by NFA A.

Thus, we have constructed an NFA A such that $L(A) = L_1^C$. This implies L_1^C is a Regular Language. Consequently, by closure of regular languages under complement, $L_1 = L(M)$ is a regular language. This implies all-NFAs recognize the class of regular languages.
Hence proved.

3 Repeat Operator Closure

Construction: Let us say we have a DFA called $A = (Q, \Sigma, q_0, F, \delta)$ corresponding to the language L. If there is no transition from q_i to q_j, for any $q_i, q_j \in Q$, we define $\delta(q_i, \phi) = q_j$. Now, we will construct an NFA called $B = (Q', \Sigma, q_0, F, \delta')$, in the following manner. Consider any transition $\delta(q_i, \lambda) = q_j$ in A, we remove this transition and add a state q_{ij} to Q such that $\delta'(q_i, \lambda) = q_{ij}$ and $\delta'(q_{ij}, \lambda) = q_j$. We repeat this process for all pairs $q_i, q_j \in Q$. We call the new states and transition sets Q' and δ'. The accepting states, start state and Σ remain the same for A and B.

Claim 3.1: Let L be a regular language corresponding to the DFA $A = (Q, \Sigma, q_0, F, \delta)$ and $R(L) = \{l_1 l_1 l_2 l_2 ... l_k l_k | l_1 l_2 .. l_k \in L, k > 0\}$, then the NFA B created by the construction described above on A accepts the language $R(L)$

Proof by Deduction:

- **Part I:** We show $R(L) \subseteq \mathcal{L}(B)$ i.e if $w \in R(L)$ then the NFA B accepts it. If $w = l_1 l_1 ... l_k l_k$, then there exists $w' = l_1 .. l_k \in L$. Consider the run in which w' is accepted by A, the DFA for L. Let the order of the states visited by this run be $q_0, q_1, q_2, ...q_k$. We know that $q_k \in F$ By definition of the construction, all these states also exist in the NFA B. Consider the path $q_0, q_{01}, q_1, q_{12}, q_2, ...q_{k-1}, q_{(k-1)k}, q_k$ in the NFA B. Clearly, this path corresponds to the word $l_1 l_1 ... l_k l_k = w$ by the definition of the construction and the fact that $\delta(q_{i-1}, l_i) = q_i$ in A. Since F is same for both states, $q_k \in F$ for B, or we have shown that the word w is accepted by B as required.
- **Part II:** We show $\mathcal{L}(B) \subseteq R(L)$ i.e if $w \in \Sigma^*$ is accepted by the NFA B, then $w \in R(L)$. Consider the path along which w is accepted by NFA B. Notice that by the construction, if at any point in the run we are at the state q_{ij}, then there is only one possible transition to q_j and the only way to arrive at q_{ij} is from q_i. Since any newly added state q_{ij} is not accepting, we can conclude the run for w on B will be of the form $s_0, s_{01}, s_1, s_{12}, s_2, ..., s_{(k-1)k}, s_k$ where $s_k \in F$ and all $s_i \in Q$. Clearly by the construction it follows that w is of the form $l_1 l_1 l_2 l_2 ... l_k l_k$. Consider the run on the word $w' = l_1 l_2 .. l_k$ in the DFA A. Clearly by the construction, in specific, by the fact that $\delta'(s_{i-1}, l_i) = s_{(i-1)i}, \delta'(s_{(i-1)i}, l_i) = s_i$ implies $\delta(s_{i-1}, l_i) = s_i$ it will follow the path $s_0, s_1, ..., s_k$ in A where $s_k \in F$. In other words, w' is accepted by the DFA A i.e $w' \in L$ and clearly `repeat(w')`$= w$. Therefore, we have shown $w \in \mathcal{L}(B)$ as required.

Using the above two claims, we can conclude that $R(L)$ is the accepting language of the NFA, B, since we have shown that $w \in R(L)$ if and only if it is accepted by B, or $\mathcal{L}(B) = R(L)$. ■

Claim 3.3: The class of regular languages is closed under the repeat operation, or equivalently for any regular language L we can say that the set $R(L) = \{l_1 l_1 l_2 l_2 ... l_k l_k | l_1 l_2 .. l_k \in L, k > 0\}$ is also a regular language.

Proof by Deduction:

Using our Construction and its correctness (Claim 3.1), we have constructed an NFA B with the accepting language $\mathcal{L}(B) = R(L)$. Hence, this is sufficient to prove that $R(L)$ is a regular language as required. This completes our proof. ■

4 Detecting Same Languages

Lemma: For DFA $D_1 = (Q_1, \sum, \delta_1, q_0^1, F_1)$ and DFA $D_2 = (Q_2, \sum, \delta_2, q_0^2, F_2)$ accepting language L_1 and L_2 respectively, the DFA for L_1/L_2 is M where $M = (Q, \sum, \delta, q_0, F)$ where $Q = \{(q_1, q_2)|q_1 \in Q_1, q_2 \in Q_2\}$, $q_0 = (q_0^1, q_0^2)$, $\delta((q_i^1, q_j^2), x) = (\delta_1(q_i^1, x), \delta_2(q_j^2, x))$ and $F = \{(q_1, q_2)|q_1 \in F_1, q_2 \notin F_2\}$.

Proof of Lemma

- Part 1 First we prove that L_1/L_2 is a subset of $L(M)$. Let x be a string belonging to L_1/L_2. This implies $x \in L_1$ and $x \in L_2$. So, from the transitions defined by δ we reach a state (q_i, q_j) such that $q_i \in F_1$(as $x \in L_1$) and $q_j \notin F_2$(as $x \notin L_2$). So, we reach a final state in M which imply $x \in L(M)$. This implies $L_1/L_2 \subset L(M)$.

- Part 2 Now we prove that $L(M)$ is a subset of L_1/L_2. Let x be a string belonging to $L(M)$. This implies x reaches one of the final states, say (q_i, q_j). This means x reaches q_i which belongs to F_1 when ran through DFA for L_1. Thus, $x \in L_1$. Similiarly, q_j does not belong to F_2 and thus, $x \notin L_2$. This implies $x \in L_1/L_2$. This implies $L(M) \subset L_1/L_2$.

Thus, we have $L(M) = L_1/L_2$.

Argument 1 To prove that DFA D_1 and DFA D_2 are equivalent, we use the fact that if $L_1 = L_2$ then $L_1/L_2 = \phi$ and $L_2/L_1 = \phi$. So, the DFA M_1 for (L_1/L_2) and DFA M_2 for (L_2/L_1) in this case must be so that the final state for this language is not reachable from the initial state.

Proof of Correctness

- We prove the above fact by contradiction. Suppose a final state is reachable for the DFA M from the starting state. Then, the path through which we reach the final state has a string that belongs to the language M and thus the language accepted by this DFA is not empty. This is a contradiction to the fact that the language is empty. Therefore, no final state is reachable from the initial state for this DFA.

Algorithm

- We first make a DFA M_1 for (L_1/L_2) and a DFA M_2 for (L_2/L_1) using the above lemma. If DFA D_1 is equivalent to DFA D_2 then $L_1 = L_2$ and from the above theorem M_1 and M_2 will not have any reachable final state. We give an algorithm to check for an automaton if it has a reachable final state.

- For a DFA, make all states as nodes and if there is a transition from state q_i to state q_j, then make a direct edge from q_i to q_j. We can ignore self loops as self loops has no contribution to the path that the automaton traverses to check whether the string belongs to the language. Let the obtained graph be G.

- Now, we start a DFS for the graph G starting from the initial state. While DFS if we don't reach any of the final state, then we come to know the language accepted by the DFA is ϕ.

Proof of correctness of algorithm

- If the algorithm is able to reach a final state starting from the initial state, then by using the alphabets in the path and joining them gives us a string that belongs to the language(this follows because we were able to reach a final state). This means the language is not empty. But if we were not able to reach a final state from the initial state, this means there is no path in the DFA to reach the final state and hence, the DFA will not accept any string and the language accepted by such an automaton is ϕ.

5 Closure under Reverse Operator

Let A be a regular language. Let M_1 be NFA corresponding to language A. $M_1 = (Q, \Sigma, \delta, q_s, q_a)$ where q_s is the single start state, and q_a is the single accept state. $\delta \subseteq Q \times (\Sigma \cup \epsilon) \times Q$ We can convert an NFA with many accept states to an equivalent NFA with single accept state by creating a dummy accept state and adding $epsilon-$transitions from the each of the original accept states to the dummy accept state.

Define NFA M_2 as $M_2 = (Q, \Sigma, \delta', q_a, q_s)$ i.e, the start state of M_2 is the accept state of M_1, and the accept state of M_2 is the start state of M_1. $\delta' \subseteq Q \times (\Sigma \cup \epsilon) \times Q$ such that $(q_i, \lambda, q_j) \in \delta' \iff (q_j, \lambda, q_i) \in \delta \ \forall q_i, q_j \in Q, \lambda \in (\Sigma \cup \epsilon)$. Thus, for any given pair of states $q_k, q_l \in Q$ if there is a path $P_1 \in \Sigma^*$ in M_1 from q_k to q_l, then there is a corresponding path $P_2 \in \Sigma^*$ in M_2 from q_l to q_k such that $P_2 = P_1^R$, and vice versa.

Therefore, for each word w accepted by M_1, there is a path P_1 from q_s to q_a in M_1. Correspondingly, there is a path $P_2 = P_1^R$ from q_a to q_s in M_2 that accepts the word w^R. This implies M_2 accepts reverse of words in M_1.

Let w' be a word $\in \Sigma^* - A$. w' is not accepted by M_1. This implies no path in M_1 corresponding to w' starting from q_s ends in state q_a.
Let us assume M_2 accepts word w'^R. This implies there exists a path P' in M_2 corresponding to word w'^R starting from q_a and ending at state q_s. This implies path P'^R is a path in M_1 corresponding to $(w'^R)^R = w'$ that starts at q_s and ends at q_a – a contradiction. Thus, there exists no path in M_2 corresponding to w'^R that starts at q_a and ends at q_s. This implies M_2 does not accept any word w where $w^R \notin A$.

Therefore, NFA M_2 accepts exactly the language A^R. This implies A_R is a Regular Language. Thus, regular languages are closed under reverse operation. Hence proved.

6 Closure under Homomorphism

Claim 6.1: If $h : \Sigma^* \to \Gamma^*$ is a homomorphism over strings then for any $w = a_1a_2...a_n \in \Sigma^*$, where $a_i \in \Sigma$ we have $h(w) = h(a_1)h(a_2)...h(a_n)$.

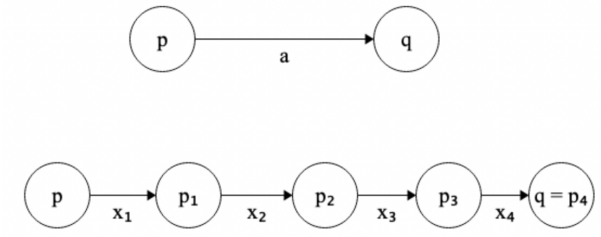
Proof by Induction:

Inductive Hypothesis: For any word $w \in \Sigma^*$ where $|w| = k$, the above claim holds.

Base Case: For $k = 0$, we have $w = \epsilon$ and by definition of homomorphisms over strings we have $h(\epsilon) = \epsilon$. Thus, trivially the base case holds.

Inductive Step: Let us say the hypothesis is true for $|w| = k$ iterations, we will extend the claim to words with $|w| = k + 1$. Let us write $w = aw'$, where $a \in \Sigma$ and $w' \in \Sigma^*$, where $|w'| = k$. By the property of homomorphisms, we have $h(w) = h(a)h(w')$. Now, using the inductive hypothesis on w' of length k, we immediately arrive at the claim for w. Thus we have shown that the inductive hypothesis holds for $|w| = k + 1$ and we are done. ■

Construction: Let us say we have a homomorphism $h : \Sigma^* \to \Gamma^*$ and a DFA called $A = (Q, \Sigma, q_0, F, \delta)$. Now, we will construct an NFA called $B = (Q', \Gamma, q_0, F, \delta')$, in the following manner. Consider any transition $\delta(p, a) = q$ in A, we remove this transition and replace it by a sequence of transitions described as $\delta(p, x_1) = p_1$, $\delta(p_1, x_2) = p_2$, $\delta(p_2, x_3) = p_3$ and so on up-to $\delta(p_{n-1}, x_n) = p_n = q$, where $h(a) = x_1x_2x_3...x_n$, $x_i \in \Gamma$. As an example, we have shown the transformation of edge in A for constructing B for $\delta(p, a) = q$ and $h(a) = x_1x_2x_3x_4$.



For every transition $\delta(p, a)$ in A, we repeat the process and add the new states into Q and remove $(p, a) \to q$ from δ and add the new transitions into it, as described above. For the start state and accepting states of B, we use q_0 and F from A, i.e, we don't change them.

Claim 6.2: Let L be a regular language corresponding to the DFA, $A = (Q, \Sigma, q_0, F, \delta)$ and $h(L) = \{h(w)|w \in L\}$, then $h(L)$ is the accepting language of the NFA, $B = (Q', \Gamma, q_0, F, \delta')$ constructed by the algorithm described above.

Proof by Deduction:

- **Part I:** We show $\underline{h(L) \subseteq \mathcal{L}(B)}$ i.e if $w \in h(L)$ then the NFA B accepts it. Let $w' = x_1x_2..x_k$ be the pre-image of w under h then $w' \in L$ and $x_i \in \Sigma$. Now, using Claim 6.1, we can say $w = h(x_1)h(x_2)...h(x_k)$. We start at the state s_0 in B and now follow along the path $p_1, p_2, ..., p_m$, where p_i's are the newly added states when we transformed $\delta(q_0, x_1)$ in A as described in the construction. After doing this, by definition of the construction, we arrive at, $p_m = s_1$ (say) $\in Q \cap Q' = Q$, where Q and Q' are the set of states of A and B. Now, we can repeat the process, by using the path added in B when transforming $\delta(s_1, x_2)$ in A. We continue this process. Now, notice that the path described in B when running on w corresponds to the path followed in A when running on w' but with extra intermediate steps, since $w' = x_1x_2...x_k$ is accepted by A, we can conclude that in the run on B will end at s_k which must the accepting state of B.
- **Part II:** We show $\underline{\mathcal{L}(B) \subseteq h(L)}$ i.e if $w \in \Gamma^*$ is accepted by the NFA B, then $w \in h(L)$. Let us call the set $N = Q' - Q$, i.e, the newly added states. Let $s_0, s_1, ..., s_m$ be the sequence in which the run on w arrives at states in $Q' - N = Q$ (recall $Q \subset Q'$), i.e, old states, that existed in the original DFA A before construction. Clearly $s_m = q_a$, i.e the accepting state and $s_0 = q_s$, i.e the staring state. Clearly between any s_i and s_{i+1}, the run arrives at a sequence of intermediate states say $p_1, p_2, .., p_{k-1}$ where $p_j \in N$. By the construction, we know that $p_1p_2..p_{k-1} = h(a)$, where $\delta(s_i, a_i) = s_{i+1}$ in the original DFA A (thus $a_i \in \Sigma$. Thus, using this fact for all s_i, s_{i+1} we can write w as $h(a_0)h(a_1)...h(a_{m-1})$ which in fact is $h(a_0a_1..a_{m-1})$. Now, note that the path $s_0, s_1, ..s_m$ is a path that is accepted by the original DFA A, hence the word $w' = a_0a_1...a_{m-1} = h^{-1}(w) \in L$. Thus since we have a pre-image $w' \in L$ of w under h, we can conclude $w \in h(L)$.

Using the above two claims, we can conclude that $h(L)$ is the accepting language of the NFA, B, since we have shown that $w \in h(L)$ if and only if it is accepted by B, or $\mathcal{L}(B) = h(L)$. ■

Claim 6.3: The class of regular languages is closed under string homomorphisms, or equivalently, for any regular language L and string homomorphism $h : \Sigma^* \to \Gamma^*$, $h(L) = \{h(w)|w \in L\}$ is also a regular language.

Proof by Deduction:

Using our Construction and its correctness (Claim 6.2), we have constructed an NFA, B with the accepting language $\mathcal{L}(B) = h(L)$. Hence, this is sufficient to prove that $h(L)$ is a regular language as required. This completes our proof. ■