CS181-Homework1

January 26th,2018

1

1.1 Exercise

a. Given DFA
$$M = \{\mathcal{Q}, \Sigma, \delta, q_0, F\}$$

According to the book,since the NFA is a generalization of DFA,we can construct our penta NFA N using the given DFA M.So in our case DFA is our NFA, our penta NFA N will reach its accept state if and only if x $\in L(M)$.

Thus, the DFA (in our case NFA) will have a single computation path and a single ending state. Therefore if the DFA(in our case NFA) reaches its accept state, then it would be accepted at the ending state. N will satisfies condition of being a penta.

b. Let NFA be

$$N = \{\mathcal{Q}, \Sigma, \delta, q_0, F\}$$

we can construct DFA
$$M = \{\mathcal{Q}', \Sigma', \delta', {q'}_0, F'\}$$

we can then set
$$Q' = \{\text{P}(\mathcal{Q})\}$$

which means every state of M is a set of states of M.

$$q_0' = \{q_0\}$$

M starts the same state corresponding to the collection containing just the start state of N.
$$F' = \{R \in \mathcal{Q}' | \text{R contains an accept state of N}\}$$

we make some appropriate changes to the five-tuple values as well as the $\epsilon - transition$.

Notice that, we are converting a DFA to NFA, but in this case, we want to convert to penta NFA N.

According the only different between the NFA that we are converting to and the penta NFA N is the at the ending accepting states, which more than one-fifth of the ending states are accepting. In this case, we need to modify our definition of F' the set of accept states

$F' = \{R \in \mathcal{Q}' | R : A \cup B$ where all elements in A are in F, elements in B are not in F and $|A| > \frac{1}{5}|5|$ $\}$

As we modified our the tuple-value for DFA, we converts the DFA to NFA with the constraints for the accepts states, and it also satisfies the condition of the penta NFA N. Therefore, when input the word w to this machine, it should behaves the same way as our penta NFA.

1.2 Exercise

Assume that we are given NFA

$$M = \{\mathcal{Q}, \Sigma, \delta, q_0, F\}$$

where L = L(M), from this point, we know that L is regular in our NFA M, we need to show L_R is also regular.

Since L_R is a set of reverse strings, we can run the machine reversely,so what that means is we can modify our existing M.

so that it runs the reverse string. To do this, we can first construct NFA by using our M and our modified NFA also satisfies L$_R = L(N)$

$$N = \{\mathcal{Q}', \Sigma', \delta', {q'}_0, F'\}$$

To do this, we need to modified a set of our states Q in M, so the accepting states in M will be the starting state Q' in N.

$$\mathcal{Q} = \{q_0, q_1, ...q_n\}$$

we can create a new state $q_(n+1)$ in N,each of the accepting states should have the epsilons in M,we can formalizes into the following Q'

$$\mathcal{Q}' = Q \cup \{q_{n+1}\}$$

$$\delta'(q_{i,j}, a) = \{q_j | \delta'(q_j, a) = q_i \text{ in which } q_{i,j} \in \mathcal{Q}\}$$

$$\delta'(q_i, a) = q_j | q_j \in F$$

$$q_0' = q_{n+1}$$

so now we have constructed our 'reverse machine', we know that NFA M recognizes L such that L(M) = L, and we also know that $L_R(N) = L_R$. We can formally show that if a word w is accepted by our M then $w_{reverse}$ is also accepted by our N.

According to the book, the definition of a NFA accepts a word w if $w = y1...ym$, where y_i is a member of Σ_ϵ and a sequence of states $(r_0, ...r_m)$ exists in $\mathcal{Q}$.

1. $r_0 = q_0$
2. $r_{i+1} \in \delta r_i, y_{y+1}$
3. $r_m \in F$

In this case, a given word w will satisfies these constraints and recognized by the M. we can add ϵ to the $w_{reverse}$ where $w_{reverse} = \epsilon y_m..y_1$. and our state list is $q_{n+1}, r_m....r_0$ by definition, N will accepts $w_{reverse}$ if M accepts w.

1.3 Exercise

Assume that we are given NFA

$$M = \{\mathcal{Q}, \Sigma, \delta, q_0, F\}$$

where L = L(M) and L is regular. In the given problem, what we can do is having a M that transitions to the odd indexed states, but also allows ϵ-transition to all the allowable even states. To do this, we will need two copies of the M in our new constructed machine. One copy will map to the odd indexed states and another copy will be the odd indexed states. Another important aspect is odd states should follow by even states and vice versa.

$$N = \{\mathcal{Q}', \Sigma', \delta', q'_0, F'\}$$

$$\mathcal{Q}' = \{q_0, odd, q_1, odd....q_n, odd, q_1, even....q_n even\}$$

The $\mathcal{Q}'$ should consist of two copies of M, a copy of even states and a copy of odd states.

$$\delta'(q_{i,j}, a)_{odd} = \{q_h, odd | \delta'(q_i, a) = q_h \text{ in which } j = even\}$$

$$\delta'(q_{i,j}, a)_{even} = \{q_h, even | \exists a \text{ s.t. } \delta'(q_i, a) = q_h \text{ in which } j = odd\}$$

Our transition function should capture the transition of both even and odd.

$$q'_0 = q_0, even$$

The start state in our machine N should be q_0 even.

$$F' = \{q_{i,j} | q_i \in F\}$$

so now we have constructed our machine N, we know that NFA M recognizes L such that L(M) = L, and we also know that $L_{alt}(N) = L_{alt}$. We can formally show that if a word w is accepted by our M then a word w is also accepted by our N.

According to the book, the definition of a NFA accepts a word w if $w = y1...ym$, where y_i is a member of Σ_ϵ and a sequence of states $(r_0, ...r_m)$ exists in $\mathcal{Q}$.

1. $r_0 = q_0$
2. $r_{i+1} \in \delta r_i, y_{y+1}$
3. $r_m \in F$

In this case, a given word w will satisfies these constraints and recognized by the M. In our new constructed machine N, w_{alt} has the form of $y1\epsilon y3\epsilon y5\epsilon$..... Our N states sequence is $\{r_0 even, r_1 odd, r_2 even, r_3 odd, r_4 even, r_5 odd......\}$, thus our new machine N will accept w_{alt} in corresponding to our machine N state sequence.

1.4 Exercise

Assume that we are given NFA

$$M = \{\mathcal{Q}, \Sigma, \delta, q_0, F\}$$

where $L = M(L)$ and L is regular, in order to prove $L_{\frac{1}{2}}$ is regular, we will have M takes the input in the forward and backward directions. if they end up in the same states, then our new machine N should recognizes the new string $L_{\frac{1}{2}}$ where $L_{\frac{1}{2}} = N(L_{\frac{1}{2}})$

$$N = \{\mathcal{Q}', \Sigma', \delta', q'_0, F'\}$$

In order to have the machine run parallel, which means one state in $\mathcal{Q}'$ will keep track of two states in $\mathcal{Q}$. To do this, we can use the property of Cartesian product of $\mathcal{Q}$ we learned in class. We will also need to start run from the ending state of M, so we will create new start that consists of (q_{start}, q_{end}) To formalize this,

$$Q' = (Q \times Q) \cup (q_{start}, q_{end})$$

notice that our original $Q = q_0, q_1, ...q_n$

we also need to modify our transition function that allows the machine to run forward and backward in parallel.

$$\delta'(q_{i,j}, a) = \{(q_h, q_l) | (\delta, a) = q_h \wedge \exists b \text{ s.t. } \delta(q_l, b) = q_j\} \text{ in which } (Q \times Q) \cup (q_{start}, q_{end})$$

$$\delta'(q_{i,j}, a) = \{(q_0, q_k) | q_k \in F\} \text{ in which } i = start \wedge j = end \wedge a = \epsilon$$

In this step, our start states q'_0 will have (q_{start}, q_{end}), the set of F' states will have the following property where each states of being two states in Q

$$F' = q_{i,j} \in Q'$$

by now, we have construct a NFA N that recognizes $L_{\frac{1}{2}}$, and we know that our NFA M recognizes L where $M(L) = L$, we can shows that if a word w is accepted by our M, then a word $w_{\frac{1}{2}}$ is also accepted by N.

According to the book, the definition of a NFA accepts a word w if $w = y1...ym$, where y_i is a member of Σ_ϵ and a sequence of states $(r_0, ...r_m)$ exists in Q.

1. $r_0 = q_0$
2. $r_{i+1} \in \delta r_i, y_{y+1}$
3. $r_m \in F$

In this case, a given word w will satisfies these constraints and recognized by the M. In our new constructed machine N, $w_{\frac{1}{2}}$ has the form of $\epsilon y1y2y3....y_{\frac{m}{2}}$. Our N states sequence is $\{q'_0, (r_0, r_m), (r_1, r_{m-1}).....(r_{\frac{m}{2}}, r_{\frac{m}{2}})\}$, thus our new machine N will accept $w_{\frac{1}{2}}$ in corresponding to our machine N state sequence.