Student ID: 004757414

Collaborators: Terry Ye

CS181 Winter 2019 – Problem Set 1

Due Tuesday, January 29, 11:59 pm

- Please write your student ID **and the names of anyone you collaborated with** in the spaces provided and attach this sheet to the front of your solutions. **Please do not include your name anywhere since the homework will be blind graded.**
- An extra credit of **5%** will be granted to solutions written using LaTeX. Here is one place where you can create latex documents for free: `https://www.overleaf.com/`. The link also has tutorials to get you started. There are several other editors you can use.
- If you are writing solutions by hand, please write your answers in a neat and readable hand-writing.
- Always explain your answers. When a proof is requested, you should provide a rigorous proof.
- 20% of the points will be given if your answer is "I don't know". However, if instead of writing "I don't know" you write things that do not make any sense, no points will be given.
- The homework is expected to take anywhere between 10 to 16 hours. You are advised to start early.
- Submit your homework online on the course webpage on Gradescope.

> Note: *All questions in the problem sets are challenging; you should not expect to know how to answer any question before trying to come up with innovative ideas and insights to tackle the question. If you want to do some practice problems before trying the questions on the problem set,* ***we suggest trying Exercise problems 1.4, 1.5, 1.9, 1.10, and 1.11 from the book. Do not turn in solutions to problems from the book.***
>
> The machines that we called "Finite State Machines" in class are also called "Deterministic Finite Automata (DFA)" and the machines we called "Magical Finite State Machines" in class are also called "Non-Deterministic Finite Automata (NFA)".

Hint on all construction problems: If you want to prove that L is regular, it suffices to give an NFA for it. On the other hand, if you are told to assume that L' is regular, this means that there must exist a DFA recognizing L'.

1 Problem1

If L is regular,let M_1 be the NFA such that M_1 recognizes L, the 5-tuple for M_1 is $(Q, \Sigma, \delta, q_0, F)$. We can construct a new NFA M_2 with

$\Sigma^{`} = \Sigma$
$q_0^{`}$ = a new state that has a ϵ connection with all states in F
$Q^{`} = Q \cup \{q_0^{`}\}$
transition function: $\delta^{`}(q, a) = \{q_1 | \delta(q_1, a) = q\}$ if q $\neq q_0^{`}$,
$\delta^{`}(q, a) = \emptyset$ if $q = q_0^{`}$ and $a \neq \epsilon$.

So this is a new NFA M_2. M_2 accepts a string if and only if the reverse of it can go from q_0 and ends in F because this NFA is moving in opposite direction to the original M_1, so M_2 recognizes L_R, which means L_R is regular.

2 Problem2

For any regular language L, let M be the NFA such that M recognizes L, the 5-tuple of M is $(Q, \Sigma, \delta, q_0, F)$. We can constaruct a new NFA $M^{`}$ with tuple $(Q^{`}, \Sigma^{`}, \delta^{`}, q_0^{`}, F^{`})$.

$Q^{`} = (Q \times Q) \cup \{q_0^{`}\}$
$\Sigma^{`} = \Sigma$
$q_0^{`}$ = a new state that has a ϵ connection with all states in $\{(q_0, f) | f \in F\}$
$F^{`} = \{(x, x)|\}$ = all stats that is a Cartesian product of two same states.
$\delta^{`}((q_1, q_2), a) = \{(\delta(q_1, a), q^{`}) | \exists b \in \Sigma$ such that $\delta(q^{`}, b) = q_2\}$

$L_{\frac{1}{2}} \subseteq L(M^{`})$
Proof: If a string x is in $L_{\frac{1}{2}}$, NFA $M^{`}$ will take the input x and the first part will act the same as in M and end at M(x). Because $x \in L_{\frac{1}{2}}$, $\exists y$ such that $|x| = |y|$ and $xy \in L$, which means M(x) is $|y|$ steps away from an accept state in F. So (M(x),M(x)) will be in the final states of $M^{`}(x)$ because $\exists$ string of size $|x|$ from M(x) to an accept state in F and the second part is able to go back from the accept state to M(x). So $M^{`}$ will accept x as (M(x),M(x))$\in F^{`}$ and $L(M^{`}) \subseteq L_{\frac{1}{2}}$

$L(M^{`}) \subseteq L_{\frac{1}{2}}$
Proof: If a string x is in $L(M^{`})$, then $M^{`}(x)$ goes to a state (q,q). By the definition of $M^{`}$, the first part will follow M and thus q = M(x). Second part means that M is able to go $|x|$ steps from one of the accept states to q. Let y be the reverse of the steps taken by M to go back from one of the accept states to q. $|x| = |y|$ and $xy \in L$, so $x \in L_{\frac{1}{2}}$, therefore $L_{\frac{1}{2}} \subseteq L(M^{`})$.

3 Problem3

For any language L(M_{2p}) such that $\forall x \in L$, x$x is recogized by a normal DFA M_{2p} with $(Q_{2p}, \Sigma_{2p}, \delta_{2p}, q_{2p}, F_{2p})$, we construct a NFA M with 5-tuple $(Q, \Sigma, \delta, q_0, F)$:

$Q = Q_{2p} \times Q_{2p} \times Q_{2p} \cup \{q_0\}$
$\Sigma = \Sigma_{2p}$ without $
$q_0 =$ a new start state
$F = \{(a, a, b)|b \in F_{2p}\}$
$\delta(q_0) = \{(q_{2p}, a, a)|a \in Q_{2p}\}$
$\delta((q_1, a, q_2)) = \{(\delta_{2p}(q_1), a, \delta_{2p}(q_2))\}$

M will start by initializing the second part to every one of Q_{2p} state. The first part is equal to M_{2p} reading x from the start state of M_{2p}. The third part is equal to M_{2p} reading x from the second part. Second part doesn't change during reading

$L(M_{2p}) \subseteq L(M)$
Proof: Suppose a string x is in L(M_{2p}) which means M_{2p} accepts $x\$x$, then there will be a state $q \in Q_{2p}$ such that q is where M_{2p} at when it reads the $ symbol. Then for the state (q_{2p}, q, q), which is connected with q_0, q_{2p} will end in q and the third will end in one of states in F, so x will be accepted by M.

$L(M) \subseteq L(M_{2p})$
Proof: Suppose a string x is in L(M), that means $\exists q$ such that $M_{2p}(x)$ will ends at q and $M_{2p}(x)$ starting at q will end in one of states in F. Then when M_{2p} reads the first pass of x, it will reach q and then it reads the second pass of q and reaches one of the accepting states. So x will be accepted by $L(M_{2p})$.