CS 4510 Automata and Complexity 9/8/2022

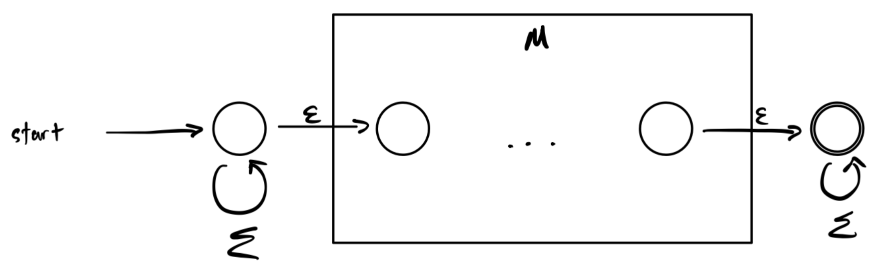
Homework 2: Nondeterminism and Pumping Lemma

Due:9/15/2022

1. **Nondeterminism** (4 points)

   (2 points - *Section A*) For any regular language L, give a NFA that accepts $L_1 = \{axb \mid x \in L, a, b \in \Sigma^*\}$, i.e. the set of all strings that contain a string from L as a substring.

Solution:



Let $M_1 = (Q_1, \Sigma, \delta_1, q_1, F_1)$ recognize L.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize L_1.

(a) $Q = Q_1 \cup q_0 \cup F$

(b) $\Sigma = \Sigma$

(c) The state q_0 is the new start state.

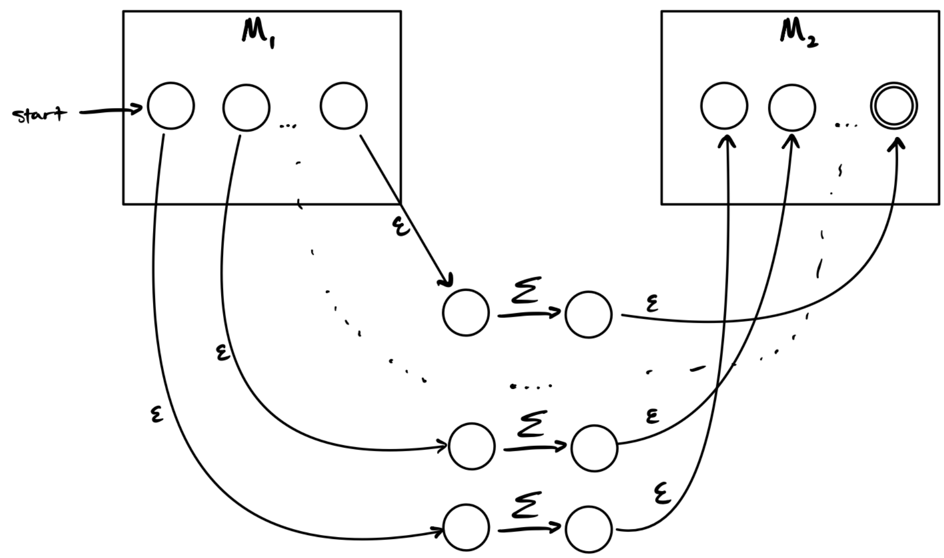
(d) Define δ so that for any $q \in Q$ and any $a \in \Sigma_\varepsilon$:

$$\delta(q,a) = \begin{cases} \delta_1(q,a) & q \in L \text{ and } a \neq \varepsilon \\ \{F\} & q \in F_1 \text{ and } a = \varepsilon \\ \{F\} & q \in F \text{ and } a \neq \varepsilon \\ \{q_0\} & q = q_0 \text{ and } a \neq \varepsilon \\ \{q_1\} & q = q_0 \text{ and } a = \varepsilon \end{cases}$$

(e) F is a new final state which is also accepting.

(2 points - *Both*) For any regular language L, prove that $L' = \{xay \mid xy \in L, a \in \Sigma, x, y \in \Sigma^*\}$ is regular, i.e. the set of all strings from which deleting exactly one character gives a string from L. For example, if L were binary palindromes (words that are the same when reversed), some words in L' would include 10010, 100, 1110001011, since deleting the red character from each string produces a palindrome.

Solution:



We create 2 copies of M, denoted as M_1 and M_2. Let $M_1 = (Q_1, \Sigma, \delta_1, q_{0_1}, F_1)$ and $M_2 = (Q_2, \Sigma, \delta_2, q_{0_2}, F_2)$ recognize L.

Construct $N = (Q, \Sigma, \delta, q_0, F)$ to recognize L'.

(a) $Q = Q_1 \cup Q_2$

For each state q_i in Q, we write q_{i_1} as the corresponding state in M_1 and q_{i_2} as the corresponding state in M_2. Let Q_1 and Q_2 be the set of states for M_1 and M_2 respectively.

(b) $\Sigma = \Sigma$

(c) $q_0 = q_{0_1}$

(d) $\delta(q,a) = \begin{cases} \delta_1(q_{i_1}, a) & q_i \in Q_1 \text{ and } a \neq \varepsilon \\ \delta_2(q_{i_2}, a) & q_i \in Q_2 \text{ and } a \neq \varepsilon \\ \{r_{i_0}\} & \text{r is a temporary state, } q_i \in Q_1 \text{ and } a = \varepsilon \\ \{r_{i_1}\} & q_i = r_{i_0} \text{ and } a \in \Sigma \\ \{q_{i_2}\} & \text{q is the corresponding state in } Q_2\text{, } q_i = r_{i_1} \text{ and } a = \varepsilon \end{cases}$

(e) $F = F_2$ (the accepting states in M_2 only).

(2 points - *Section X*) Let L be a regular language, and let $L^{\#}$ be the set $\{x \in \Sigma^* \mid \text{for some } y \in L, y \text{ has the same number of 1's as x}\}$. Prove that, if L is regular, $L^{\#}$ is regular.

2. **Regular Expressions** ***Both Sections***(4 points)

(2 points) Give a regular expression for each of the following languages:

- The set of all strings with an even number of 1's.
- The set of all even length strings with at most two 0's

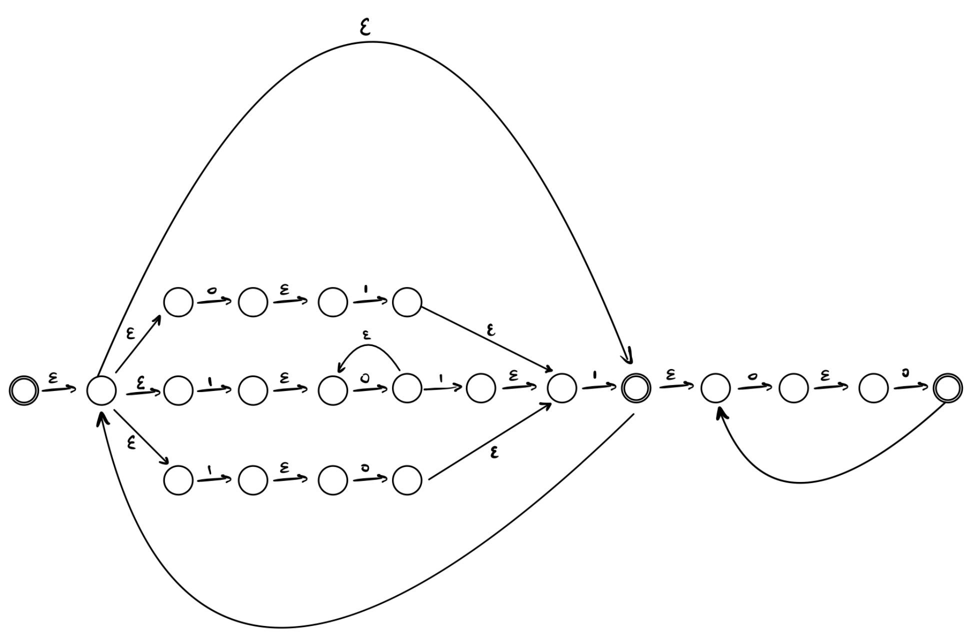
Solution:

- $0^*(10^*1)^*0^*$
- $(11)^* \cup (11)^*10(11)^* \cup (11)^*01(11)^* \cup (11)^*0(11)^*0(11)^* \cup (11)^*10(11)^*01(11)^* \cup (11)^*10(11)^*10(11)^* \cup (11)^*01(11)^*01(11)^* \cup (11)^*01(11)^*10(11)^*$

(2 points) Give an equivalent NFA for the following regular expression:

$((01 \cup 10^*1 \cup 10)1)^*(00)^*$

Solution:



3. **Pumping Lemma** (4 points)

(*Section A*) Prove that $L_1 = \{a^i b^j \mid |i-j| \text{ is prime}\}$ is not regular.

Proof. by contradiction.

Assume that L_1 is regular. Let p be the pumping length given by the pumping lemma. Choose s to be the string $a^p b^{p-k} \in L_1$ where k is prime and $k >= 2$ since 2 is the smallest prime. Because s is a member of L_1 and s has length more than p, the pumping lemma guarantees that s can be split into three pieces, $s = xyz$, where for any $i \geq 0$ the string $xy^i z$ is in L_1. Take $x = a^{p-1}, y = a, z = b^{p-k}$. We consider the following case to show that this result is impossible.

The string y consists only of only the letter a. In this case, the string $xy^k z$ now has k more a's than letter b's, specifically $a^{p+k} b^{p-k}$ which is $\notin L_1$ since $(x+k) - (x-k) = 2k$ which can never be prime.

$\square$

(*Section X*) Prove that $L_2 = \{a^i b^j \mid i, j \text{ are relatively prime}\}$ is not regular.

4. **Pumping Lemma Adjacent** ***Section A only*** (4 points)

A *minimal* DFA D for a language L is a DFA such that any DFA for L has at least as many states as D. Complete the steps below to prove the following claim:

Claim: For any positive integer $n \geq 3$, there is a language whose minimal DFA contains n states.

Proof: For some $k \in \mathbb{Z}$, let $L_k = \{a^k\}$, a set with one element. Now, suppose D is a minimal DFA for L_k. You will prove that D has at least $k+2$ states. Let $\{q_i\}_{i=0}^k = q_0, \ldots, q_k$ be the sequence of states (not necessarily distinct!) that D follows when reading a^k.

(1 points) Explain why $\{q_i\}_{i=0}^k$ cannot contain any repeated states (hint: the proof of the pumping lemma), and conclude that D has at least $k+1$ states.

Solution:
Proof by contradiction.

Assume that $q_i = q_j$ for 2 states in $\{q_i\}_{i=0}^k = q_0, ..., q_k$ such that $0 \leq i < j \leq k$. We also know that q_k must be an accepting state to accept string a^k which contains exactly k a's. Because $q_i = q_j$, we have a loop of $j - i$ a's. In a scenario where we traverse this loop 2 times, the DFA D can also accept a string with $k + (j - i)$ a's. This means that DFA D also accepts the string $a^{k+(j-i)}$.

This is a contradiction. Our assumption that 2 states $q_i = q_j$ in D from $q_0 \cdots q_k$ are the same is wrong, so all states in $q_0 \cdots q_k$ must be distinct. If all states in $q_0 \cdots q_k$ are distinct then D must have $k - 0 + 1 = k + 1$ states to accept only the string a^k.

(1 point) Now consider the state that D ends in upon reading the string a^{k+1}, and argue that D must have at least $k+2$ states.

Solution:

To guarantee that D accepts only an input string a^k with k a's, there must only be one way to reach the accepting state q_k.

Let q_{k+1} be the state that D ends in when reading the string a^{k+1}. q_{k+1} must be a "trash state" that can never reach q_k because in part (a), we showed that states $q_0 \cdots q_k$ must be distinct. If q_{k+1} is the same state as one of $q_0 \cdots q_k$, then there would be a way where q_{k+1} to be able to transition to q_k again due to the loop upon reading more a's and accepting a string with more than k a's. This is a contradiction with our requirements for the DFA because we only want to accept a string with exactly k a's.

Remember in part (a), we found that there are k+1 distinct states for $q_0 \cdots q_k$. Now upon adding q_{k+1}, the "trash state", we now have a total of $k+2$ distinct states in D.

Additionally, we could have multiple "trash states" $q_{k+1}, q_{k+2}, q_{k+3}, etc.$ which all do not reach q_k. This means that D can have *at least* $k+2$ states. However, we can combine all of these "trash states" into one "trash state" so that DFA D has exactly $k+2$ states.

(1 point) Demonstrate that D has exactly $k+2$ states by giving an explicit construction of D.

Solution:

Let $D = (Q, \Sigma, \delta, q, F)$.

(a) $Q = \{q_i \mid 0 \leq i \leq k+1\}$

(b) $\Sigma = \{a\}$

(c) $\delta(q, a) = \begin{cases} q_{i+1} & 0 \leq i \leq k \\ q_{k+1} & i > k \end{cases}$

(d) $q = q_0$

(e) $F = q_k$

(1 point) Complete the conclusion of this proof: "Therefore, we have given a constructive proof of the original claim. For any $n \in \mathbb{Z}, n \geq 3$, the language ___________ has a minimal DFA with n states."

Solution:

$L_{n-2} = \{a^{n-2}\}$

Note that in the previous portions of this problem, we showed that $L_k = \{a^k\}$ has $k+2$ states. That means that L_{n-2} will have $(k+2) - 2 = k$ states.

5. **Polynomial Length** ***Section X only*** (4 points)
Given $f(n)$, a polynomial with non-negative integral coefficients, let $L_f = \{1^{f(n)} \mid n \in \mathbb{N}\}$.

- For exactly which $f(n)$ is L_f regular? Prove your answer.
- Prove that for all other $f(n)$, L_f is not regular.