COMS 331: Theory of Computing, Fall 2021
Midterm Exam

8:50AM-10:00PM, Friday, October 15, on Gradesccope.

You have TWO hours to complete and submit this exam. There are five problems, worth 100 points for the last question, and 50 points for each of the others. Any problem that you leave blank or indicate clearly that you do not want graded will receive 30% credit. Every other problem will be graded and receive 0-50 points (first four questions) or 0-25 (each part of the last question). All answers should be explained, at least briefly.

While taking the exam you MAY

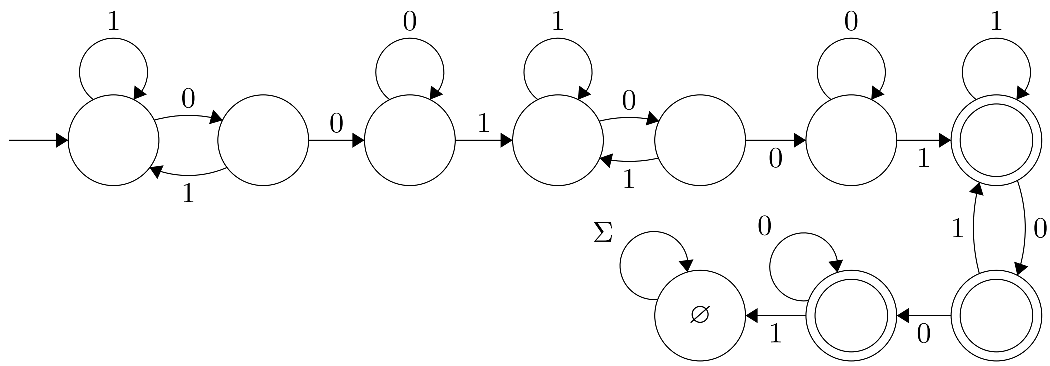
- consult your own notes and the lecture notes;
- consult the textbook and the class lectures;
- contact the instructor or TA with questions about the exam. (We'll try to answer promptly, but cannot guarantee that.)

While taking the exam you may NOT

- post questions or any information about the exam;
- discuss the exam with anyone except the instructor and TAs;
- work with anyone else or access any other resources or websites.

Problem 1. (50 points) Let A be the set of all strings $x \in \{0,1\}^*$ such that the string 001 (with these three bits consecutive) occurs in exactly two places in x. (Examples: $\underline{001}\,\underline{001} \in A$ and $1\underline{001}01\underline{001}0 \in A$, but $001 \notin A$ and $10010010010 \notin A$.)
Design a DFA M such that $L(M) = A$.

Solution



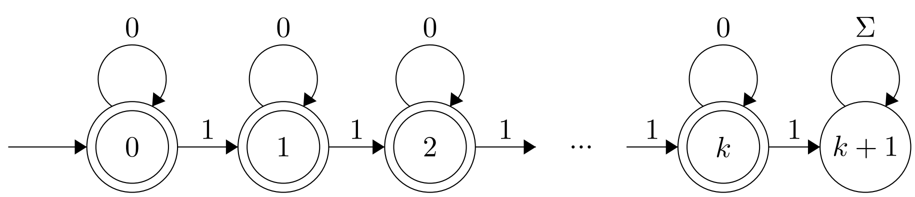
The first substring 001 will reach the 4th state, second substring 001 will reach the last state, third substring will reach the trash state. 7th to 9th states are accepting states since there are only 2 001 substrings.

Problem 2. (50 points) Prove: For every $k \in \mathbb{N}$, the language

$$A_k = \{x \in \{0,1\}^* \mid \#(1,x) \leq k\}$$

is regular, where $\#(1,x)$ is the number of 1's in x.

Solution Constructive proof: For each k, we can build a TM M for A_k where $M = (Q, \Sigma, \delta, s, F)$, $Q = \{n \mid n \in \mathbb{N}, n \leq k+1\}$, $\Sigma = \{0,1\}$, and its transitions will look as following.



k is a set number for language A_k, hence this DFA will have finite number of states. Reading in symbol 0 will simply loop on each state since 0 does not affect the membership of the string. Each state n will transition to state $n+1$ upon reading a symbol 1, except state $k+1$, which will act as a trash state. Thus this DFA accepts all strings with $\#(1,x) \leq k$ and reject all the other strings, which is A_k. Since this DFA clearly describes A_k, A_k is a regular language.

Myhill Nerode proof: There are $k+2$ equivalence classes of $\equiv_{A_k}$. For each $m, n \in \mathbb{N}$, $m < n \leq k$, $x \not\equiv y$ if $\#(1,x) = m$ and $\#(1,y) = n$ because if $z = 1^{k-n}$ then $xz \notin A_k$ but $yz \in A_k$. So here we have $k+1$ different equivalence classes. All the other strings x where $\#(1,x) > k$ belong in one equivalence class since $\forall z \in \Sigma^*$, $xz \notin A_k$.

Common mistake! A_k is not a finite language, because its strings can have any number of 0's and thus the possibility is infinite.

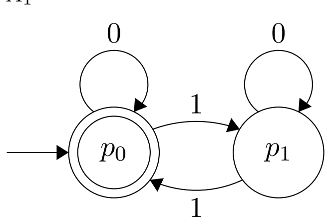
Problem 3. (50 points) Let

$$A = \{x \in \{0,1\}^* \mid \#(1,x) \text{ is even if and only if } |x| \text{ is a multiple of } 3\}.$$

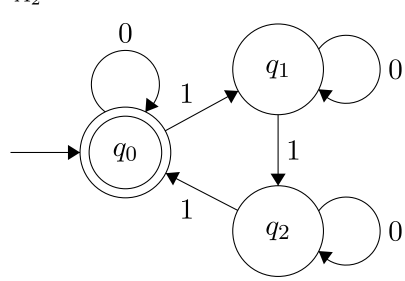
(Recall that P if and only if Q means that P and Q are both true or both false.)
Design a DFA M such that $L(M) = A$.

Solution The easiest way to solve this is to combine the DFAs for $A_1 = \{x \mid \#(1,x) \text{ is even}\}$ and $A_2 = \{x \mid |x| \text{ is a multiple of } 3\}$ using product construction, where $\delta(pq,a) = \delta_{A_1}(p,a)\delta_{A_2}(q,a)$ and $pq \in Q, p \in Q_{A_1}, q \in Q_{A_2}$.

δ_{A_1} :

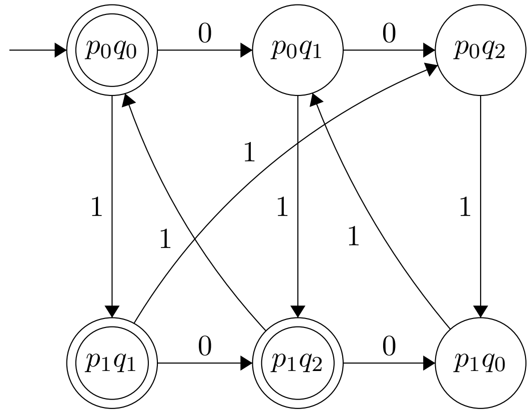


δ_{A_2} :



Since the condition here is iff, meaning to accept a string if it satisfies both condition, or none of the condition, the final state will be $F = (F_{A_1} \times F_{A_2}) \cup ((Q_{A_1} - F_{A_1}) \times (Q_{A_2} - F_{A_2}))$.

δ :



Problem 4. (50 points) Prove: For every $n \in \mathbb{N}$, there is a regular language $A \subseteq \{0,1\}^*$ such that A is not decided by any n-state DFA.

Solution For every $n \in \mathbb{N}$, define a finite (thus regular) language $A_n = \{x \in \{0,1\}^* \mid |x| \leq n\}$. We will show that A_n cannot be decided by any n-state DFA.

Assume that there is an n-state DFA that decides A_n. Since there are only n states, then there must exist 2 strings x, y such that $0 \leq |x| < |y| \leq n$ where $\hat{\delta}(s,x) = \hat{\delta}(s,y)$. Let z be some string where $|z| = n - |x|$, then $xz \in A_n$ and $yz \notin A_n$, in other words $\hat{\delta}(\hat{\delta}(s,x),z)$ should be an accept state and $\hat{\delta}(\hat{\delta}(s,y),z)$ should be non-accepting state. However, we have established that $\hat{\delta}(s,x) = \hat{\delta}(s,y)$, thus $\hat{\delta}(\hat{\delta}(s,x),z) = \hat{\delta}(\hat{\delta}(s,y),z)$, which is a contradiction.

Problem 5. (100 points) In each of the following, either give an example of languages A and B with the indicated property or state that no such languages A and B exist.

(a) A is regular and B is not regular.
Solution $A = \Sigma^*$, $B = \{0^n1^n \mid n \in \mathbb{N}\}$

(b) A is regular, $B \subseteq A$, and B is not regular.
Solution same as (a)

(c) A is regular, B is regular, and $A \cap B$ is not regular.
Solution not possible, by closure properties of regular language, the intersection of 2 regular languages has to be regular

(d) A is regular, B is not regular, and $A \cap B$ is regular.
Solution $A = \emptyset$, $B = \{0^n1^n \mid n \in \mathbb{N}\}$, $A \cap B = A$