Introduction to the Theory of Computation 2020 — Midterm 1

Solutions

Problem 1 (25 pts). Consider the language $E = \{1\}$ and the alphabet $\Sigma = \{0, 1\}$. (Note: $\Sigma = \{0, 1\}$ but not $\{1\}$.)

(a) (5 pts) Construct a DFA to recognize E. The number of states should be ≤ 3.

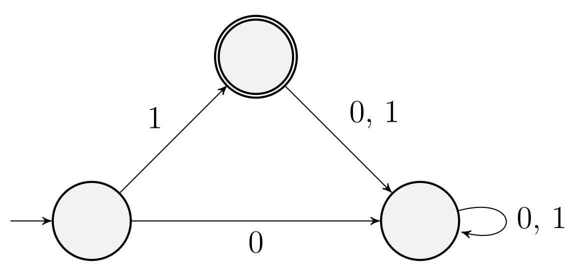
(b) (10 pts) Prove that no DFA with ≤ 2 states can recognize E. **Your proof must be clear and rigorous.**

(c) (5 pts) Construct an NFA to recognize E. The number of states should be ≤ 2. Give the formal definition of this NFA.

(d) (5 pts) Convert the NFA in (c) to a DFA by following the procedure in Theorem 1.39 of the textbook. Can this diagram be simplified to the diagram in (a)?

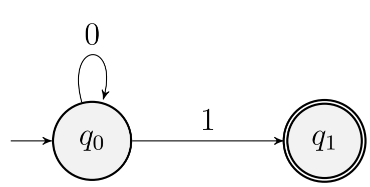
Solution.

(a) A DFA recognizing E is shown below.



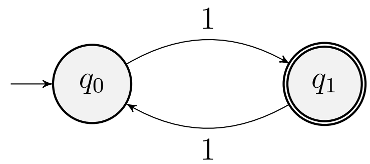
(b) As the DFA cannot accept ϵ, the start state cannot be an accept state. On the other hand, in order to accept 1, there must be (at least) one accept state. Thus the DFA cannot have only a single state. So far the DFA should be like

To accept 1 and reject 0, the two transitions from q_0 must be the following.



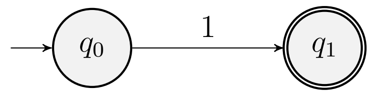
But then 01 is accepted.

Alternatively, to accept 1 and reject 11, the 1-transitions from the two states must be the following.



But then 111 is accepted.

(c) An NFA recognizing E is shown below.

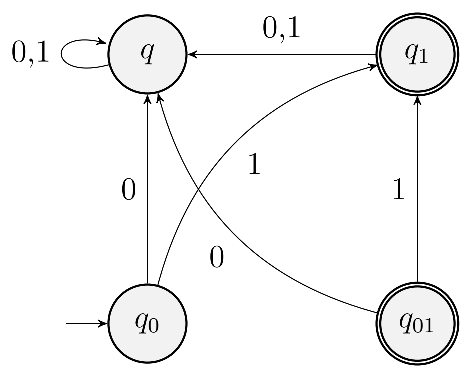


A formal definition of this NFA is $(\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$, where δ is given as

	0	1	ϵ
q_0	$\emptyset$	$\{q_1\}$	$\emptyset$
q_1	$\emptyset$	$\emptyset$	$\emptyset$

Common mistake: The transition function of an NFA is $\delta : Q \times \Sigma_\epsilon \to \mathcal{P}(Q)$, so each entry in the table should be a subset of Q. Many write q_1 instead of $\{q_1\}$.

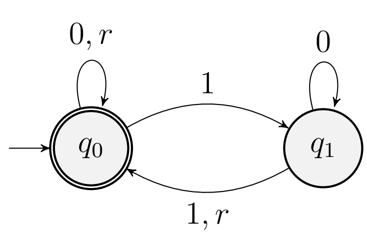
(d) The resulting DFA is shown below.



We see that the DFA in (d) is equivalent to the one in (a) by removing q_{01}, as q_{01} is not reachable from the start state.

You cannot just say simplification to (a) is possible without showing the above four-state diagram.

Problem 2 (25 pts). Consider the following DFA with $\Sigma = \{0, 1, r\}$.



(a) (5 pts) Transform the DFA to a GNFA G, and give the formal definition of G.

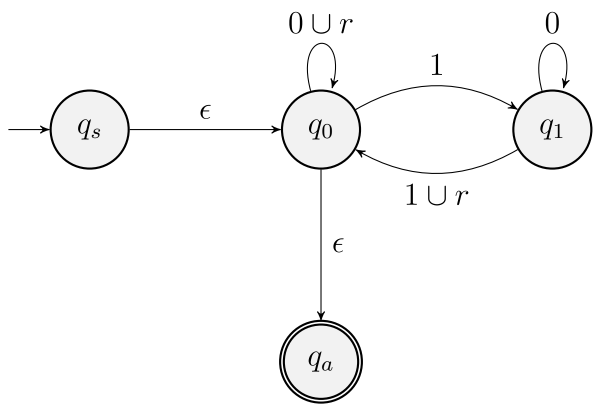
(b) (5 pts) Sequentially remove q_0, q_1 to generate a regular expression.

(c) (5 pts) Sequentially remove q_1, q_0 to generate a regular expression.

(d) (10 pts) Explain that the regular expression obtained in (c) leads to strings whose sum of all digits after the last (rightmost) r is even. **Your explanations must be clearly written.**

Solution.

(a) The following is G. Transitions with $\emptyset$ are omitted.

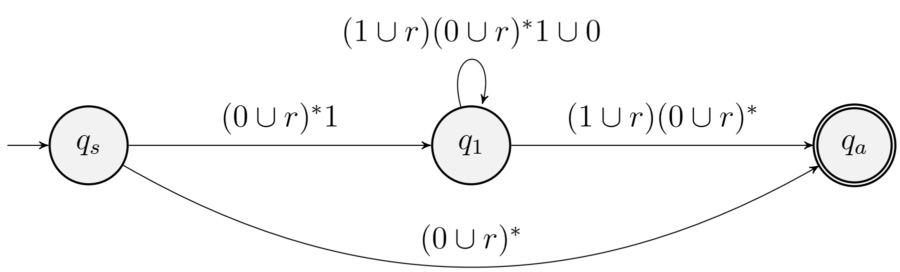


The formal definition is $G = \langle \{q_s, q_0, q_1, q_a\}, \Sigma, \delta, q_s, q_a \rangle$, where δ is

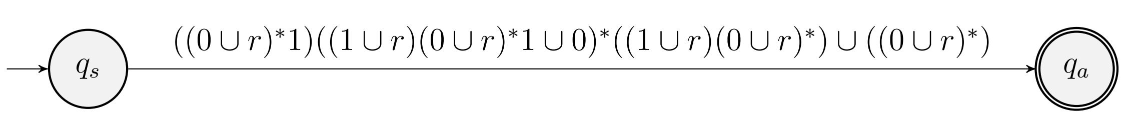
	q_0	q_1	q_a
q_s	ϵ	$\emptyset$	$\emptyset$
q_0	$0 \cup r$	1	ϵ
q_1	$1 \cup r$	0	$\emptyset$

Common mistake: For GNFA, we have $\delta : (Q - \{q_{\text{accept}}\}) \times (Q - \{q_{\text{start}}\}) \to \mathcal{R}$, but many students wrongly consider $\delta : (Q - \{q_{\text{accept}}\}) \times \Sigma_\epsilon \to Q$.

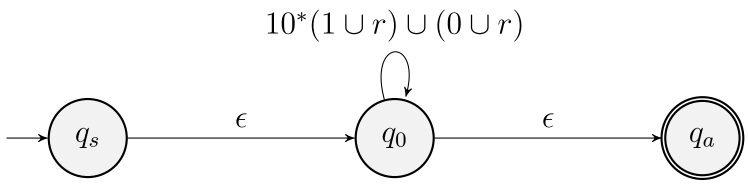
(b) After removing q_0:



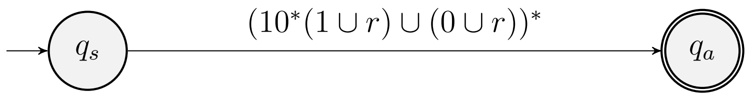
After removing q_1:



(c) After removing q_1:



After removing q_0:

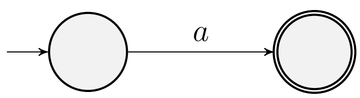


(d) For any string described by the regular expression $(10^*(1 \cup r) \cup 0 \cup r)^*$, because the expression inside the outer $()^*$ is equivalent to $10^*1 \cup 10^*r \cup 0 \cup r$, the digits after the last r must be in the form of $(10^*1 \cup 0)^*$. As both 10^*1 and 0 sum to an even number (0 and 2, respectively), any number of combinations of these two must sum to an even number as well.

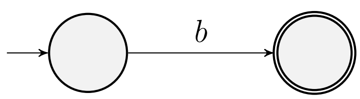
Problem 3 (20 pts). Consider $\Sigma = \{1\}$ and the following regular expression.

$$\emptyset \circ (1^*) \tag{1}$$

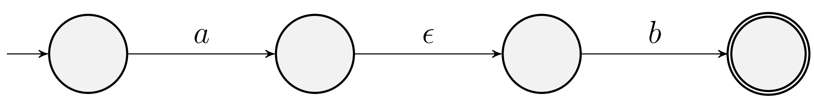
(a) (5 pts) Recall that in the textbook and slides, to convert the regular expression $(ab \cup a)^*$ to an NFA, we use the following construction. Use the same method to construct an NFA for (1).



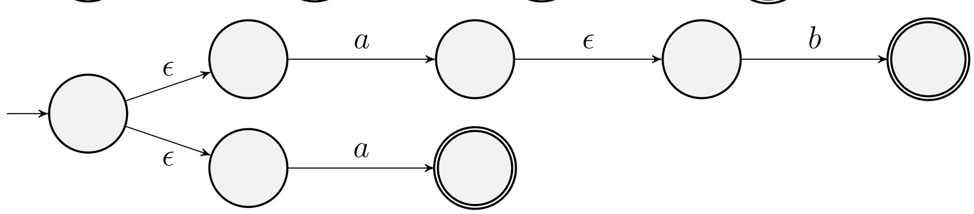
a



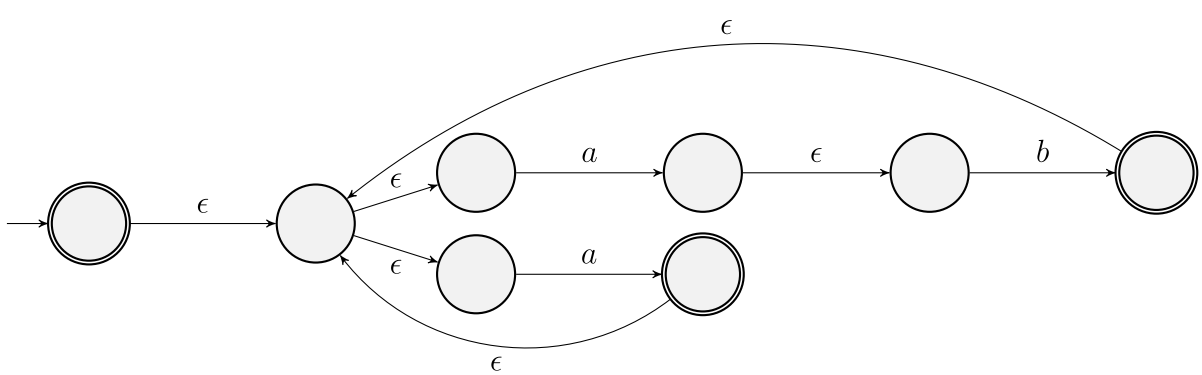
b



ab



$ab \cup a$



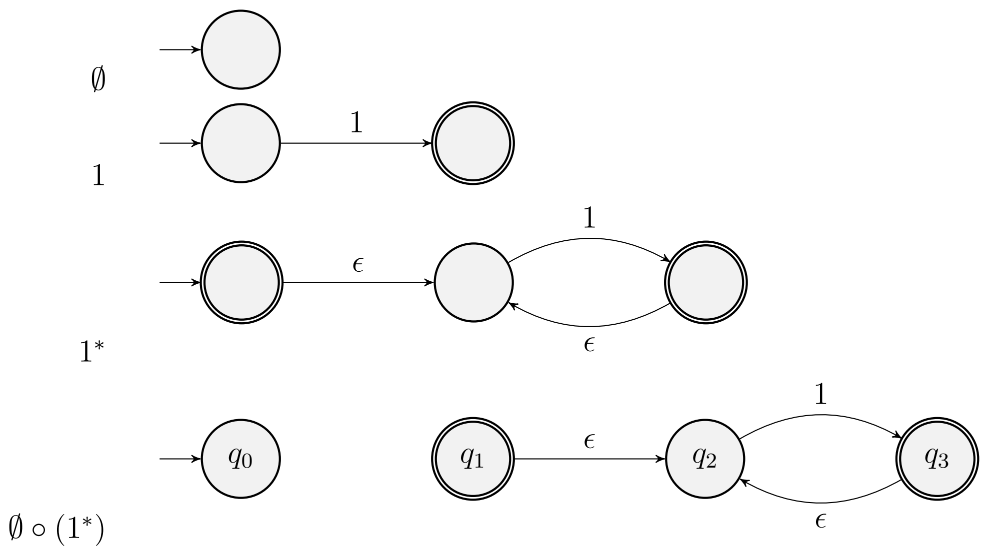
$(ab \cup a)^*$

(b) (10 pts) Convert this NFA to a DFA by the procedure taught in the class. Hint: try to think about how the graph may look like before going to draw it.

(c) (5 pts) Simplify the result in (b).

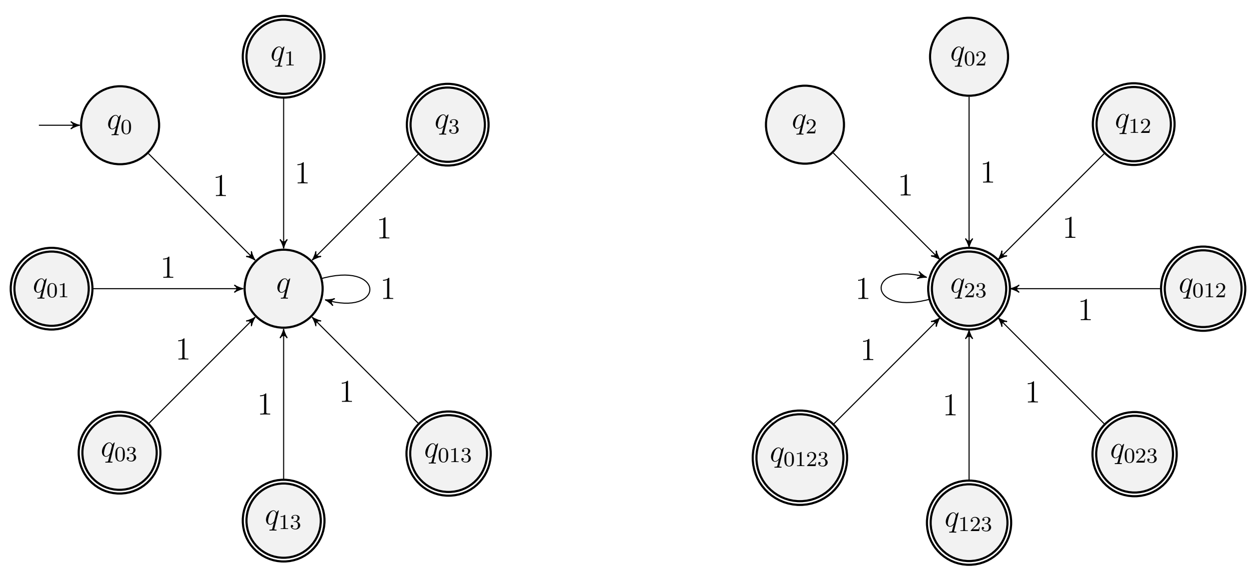
Solution.

(a) See below.



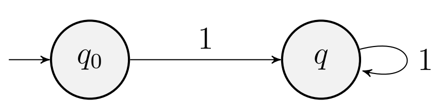
Common mistake: You cannot then simplify the diagram for (b).

(b) The resulting DFA is shown below.



Some students wrongly consider $\Sigma = \{0, 1\}$ and the diagram becomes complicated. You should always read the problem statement carefully.

(c) Because only q and q_0 are reachable from the start state, the DFA can be simplified to



Problem 4 (30 pts). Determine and prove whether the following languages are regular or not.

(a) (10 pts) $\Sigma = \{0,1\}, A = \{0^i 1^j \mid i > 3j \text{ and } i, j \geq 0\}$.

(b) (10 pts) $\Sigma = \{0\}, B = \{0^n \mid n \text{ is prime}\}$.

(c) (5 pts) $\Sigma = \{0,1\}, C = \bigcup_{i=20}^{2020}\{0^i 1^i\}$.

(d) (5 pts) $\Sigma = \{0,1\}, D = \bigcup_{i=20}^{\infty}\{0^i 1^i\}$.

Solution.

(a) Assume A is regular. Let p be the pumping length, and take $s = 0^{3p+1}1^p \in A$. Then $|s| \geq p$ and the pumping lemma guarantees that $s = xyz$ where $|y| > 0$, $|xy| \leq p$, and $xy^iz \in A, \forall i \geq 0$.

Because $|xy| \leq p$, the whole y must lie in 0^{3p+1}. Because $|y| > 0$, the string xy^0z must be 0^q1^p where $q < 3p+1$, i.e. $q \leq 3p$. This means $xy^0z \notin A$, a contradiction. Hence A is nonregular.

(b) Assume B is regular. Let p be the pumping length, and take $s = 0^n \in B$ where n is the smallest prime larger than p. Then $|s| \geq p$ and the pumping lemma guarantees that $s = xyz$ where $|y| > 0$, $|xy| \leq p$, and $xy^iz \in B, \forall i \geq 0$.

Consider $xy^{n+1}z$. Its length is $|xy^{n+1}z| = |xyz| + |y^n| = n + n|y| = n(|y|+1)$, which is composite as $|y| > 0$ and n is prime. This means $xy^{n+1}z \notin B$, a contradiction. Hence B is nonregular.

Common mistake: Many choose $s = 0^p$ but p may not be prime and s may not be in B.

(c) In the following we prove that each $\{0^i1^i\}$ is regular, and regular languages are closed under finite union. Since $C = \Big(\cdots((\{0^{20}1^{20}\} \cup \{0^{21}1^{21}\}) \cup \{0^{22}1^{22}\}) \cup \cdots\Big) \cup \{0^{2020}1^{2020}\}$ is a finite union, it is regular.

To be specific, to show that each $\{0^i1^i\}$ is regular, let $A_i = \{0^i1^i\}$, for $i = 20, \ldots, 2020$. When k is given, A_k is regular as it is recognized by $(Q = \{q_{00}, q_{10}, \ldots, q_{k0}, q_{11}, q_{21} \ldots, q_{k1}\}, \Sigma, \delta_k, q_{00}, \{q_{k1}\})$, where

$$\delta_k = \begin{cases} \delta_k(q_{j0}, 0) = \{q_{(j+1)0}\} & j = 0, \ldots, k-1 \\ \delta_k(q_{j0}, 1) = \emptyset & j = 0, \ldots, k-1 \\ \delta_k(q_{k0}, 1) = \{q_{11}\} & \\ \delta_k(q_{k0}, 0) = \emptyset & \\ \delta_k(q_{j1}, 1) = \{q_{(j+1)1}\} & j = 1, \ldots, k-1 \\ \delta_k(q_{j1}, 0) = \emptyset & j = 1, \ldots, k-1 \\ \delta_k(q, \epsilon) = \emptyset & q \in Q \end{cases}.$$

By Thm 1.45, we know that $A_{20} \cup A_{21}$ is regular, and denote it as B_{21}. Let us take $B_k = B_{k-1} \cup A_k$. By Thm 1.45 again, B_k is regular, for $k = 22, \ldots, 2020$. Since $B_{2020} = \bigcup_{i=20}^{2020}\{0^i1^i\}$, we have done the proof.

(d) Assume D is regular. Let p be the pumping length, and take $s = 0^q1^q \in D$ where $q = \max(p, 20)$. Then $|s| \geq p$ and the pumping lemma guarantees that $s = xyz$ where $|y| > 0$, $|xy| \leq p$, and $xy^iz \in D, \forall i \geq 0$.

Because $|xy| \leq p$, the whole y must lie in 0^q. Because $|y| > 0$, the string xy^2z must be 0^r1^q where $r > q$. This means $xy^2z \notin D$, a contradiction. Hence D is nonregular.

Common mistake: Many just choose $s = 0^p1^p$ without checking if $s \in D$. We need $s = 0^q1^q$ where (at least) $q = \max(p, 20)$ to ensure $s \in D$.