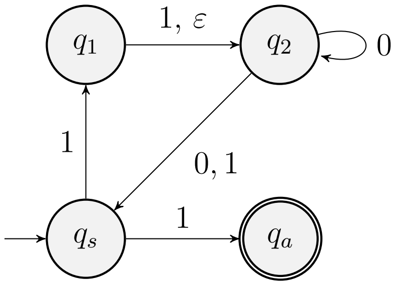
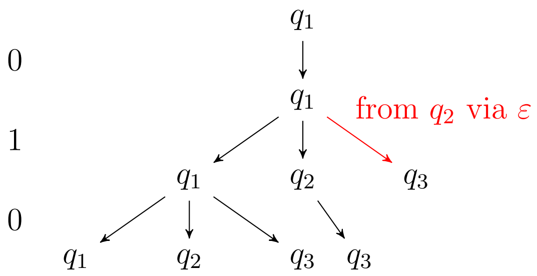
Introduction to the Theory of Computation 2023 — Midterm 1

Solutions

Problem 1 (15 pts). Let $\Sigma = \{0, 1\}$. Consider the following NFA



(a) (5 pts) Please draw the computation of this NFA on inputs "111" and "1011" and conclude whether they are accepted or rejected. A computation figure is like the following (copied from Figure 1.29 in the textbook, as an illustration and not related to the NFA in this subproblem.)



Please follow the rules in the textbook. Note that you have to list all possible states that can be reached once processing each input character.

(b) (10 pts) After Xiao-Ming finished the computation for the previous subproblem, he found that it's annoying to manually find which states can be reached with given a current state and an input character. Therefore, he wrote a subroutine for it and here is the pseudocode:

```
procedure FindReachableState(cur_state, char)
    next_level_state = $\delta$(cur_state, char)
    for state $\in$ $\delta$(cur_state, char) do
        next_level_state $\leftarrow$ next_level_state $\cup$ $\delta$(state, $\varepsilon$)
    return next_level_state
```

TA found the procedure failed in some cases. Please give a counterexample with no more than 3 states to Xiao-Ming. Note that we keep having $\Sigma = \{0, 1\}$.

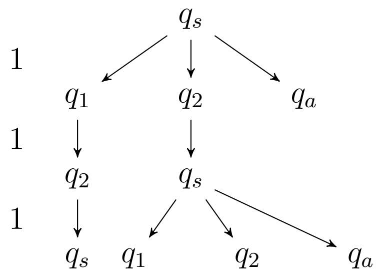
To rigourously show that your NFA is a counterexample, first give a string s in the language and then show that

- s is not accepted by Xiao-Ming's procedure
- s is accepted by the correct procedure.

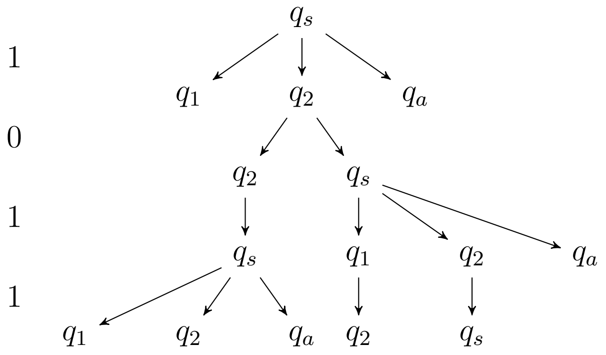
To do this you need to draw two trees like in (a).

Solution.

(a) For input "111":

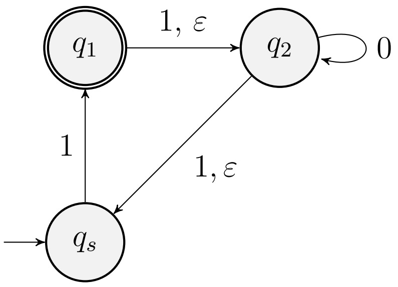


For input "1011":

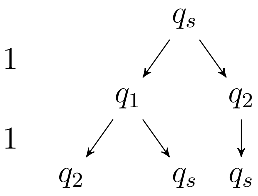


Therefore, both strings are accepted.

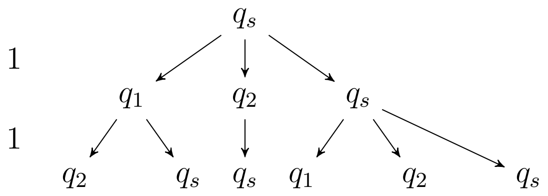
(b) Consider computing the string "11" on the 3-state NFA



If we follow Xiao-Ming's procedure, we have



So “11” is rejected. However if we follow the correct procedure, we have



Then it shows that “11” should be accepted.

Let us check the bug in the subroutine. When a new state S is added into `next_level_state` in line 4, we should recursively check whether S can bring us to other unexplored states via the ε link. Otherwise, you may miss some reachable states. For example, given current state q_s and the input character “1”, the subroutine only returns $\{q_1, q_2\}$, but the complete reachable states should be $\{q_1, q_2, q_s\}$. If q_s is not in the reachable state, you will miss the chance going back to q_s. Then, you cannot head for the accepting state q_1 when the next input character is another “1”.

Problem 2 (40 pts). Let $\Sigma = \{0, 1, 2\}$. Consider the language

$$L_1 = \{\boldsymbol{w} = w_1 w_2 \dots w_n \mid (w_1 + w_2 + \cdots + w_{n-1}) \mod 3 = w_n\}$$

For example,

$$\boldsymbol{w} = 01211 \in L,$$

since $(0 + 1 + 2 + 1) \mod 3 = 1$.

(a) (10 pts) Design an NFA recognizing L_1 using ≤ 4 states. Draw the diagram and give its formal definition. Run the string

$$01211$$

by drawing a tree like in Problem 1.

(b) (10 pts) Apply the procedure described in Theorem 1.39 in the textbook to convert the NFA in the previous subproblem into a DFA. For simplicity, please remove useless states and only show the final result.
Hint: Don’t draw all states at once. Draw states of single element subsets and gradually expand the graph.

(c) (10 pts) Now let $\Sigma = \{0, 1, 2, 3, \dots, 9\}$. Here’s the formal definition of a DFA with 20 states recognizing

$$L_1' = \{\boldsymbol{w} = w_1 w_2 \dots w_n \mid (w_1 + w_2 + \cdots + w_{n-1}) \bmod 10 = w_n\}.$$

- $Q = \{q_0, q_1, \dots, q_9, q_0', q_1', \dots, q_9'\}$ is the set of the states.
- $\Sigma = \{0, 1, \dots, 9\}$ is the alphabet.
- q_0 is the start state.
- $F = \{q_0', q_1', \dots, q_9'\}$ is the set of accept states.
- The transition function δ is given as following. Let $k = (i + j) \bmod 10$.

$$\delta(q_i, j) = \begin{cases} q_k' & \text{if } i = j \\ q_k & \text{otherwise} \end{cases} \qquad \text{for each } i = 0, \dots, 9 \text{ and } j = 0, \dots, 9$$

$$\delta(q_i', j) = \begin{cases} q_k' & \text{if } i = j \\ q_k & \text{otherwise} \end{cases} \qquad \text{for each } i = 0, \dots, 9 \text{ and } j = 0, \dots, 9$$

Please remove at least 5 useless states in the DFA and briefly explain your idea. You don't need to draw a 20-state diagram. Instead, explain why those states can be removed.

Hint: check the DFA obtained in (b) and the difference between L_1 and L_1'.

(d) (10 pts) Let $\Sigma = \{0, 1, 2\}$. Consider the language

$$L_2 = \{\boldsymbol{w} = ab \mid |\boldsymbol{w}| \geq 1,\ a \in \Sigma^*,\ b \in \Sigma^* \text{ and } (\text{sum}(a) - \text{sum}(b)) \bmod 3 = 0\}.$$

That is, each $\boldsymbol{w} \in L_2$ can be split into two parts with the same summation after taking modulo of 3. For example, the string

$$\text{“1122012”} \in L_2,$$

can be split into “11220” and “12”, and we have

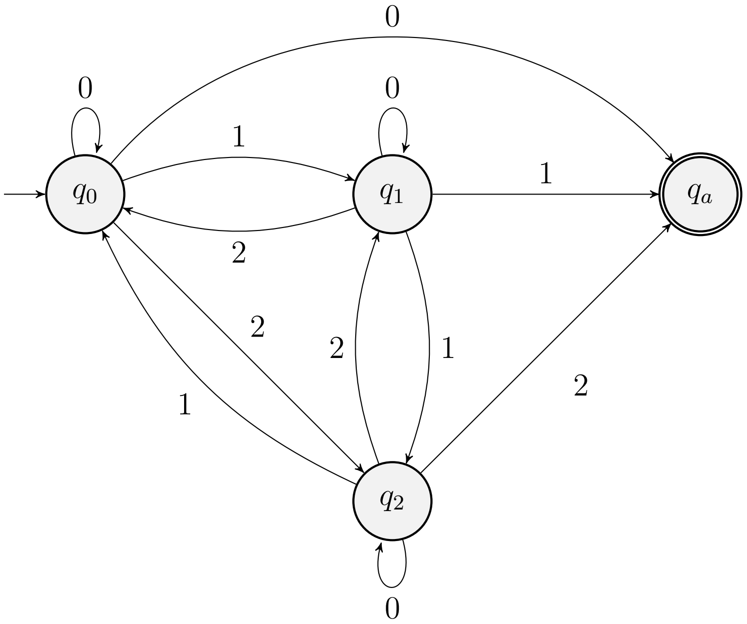
$$((1+1+2+2+0) - (1+2)) \mod 3 = 0.$$

Note that $\varepsilon \notin L_2$ but “0” $\in L_2$. Please design an NFA recognizing L_2 with no more than 6 states. For easy grading, if possible, please name your states $q_1, q_2, q_3, q_1', q_2'$ and q_3'

Hint: think from the viewpoint of $(sum(a) - sum(b)) = w_1 + \cdots + w_i - w_{i+1} - \cdots - w_n$.)

Solution.

(a) L_1 can be recognized by the following NFA



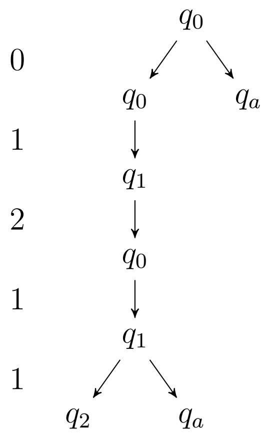
The NFA can be written as $N = (Q, \Sigma, \delta, q_0, F)$, where

$$Q = \{q_0, q_1, q_2, q_a\}$$

$$\delta = \begin{array}{c|cccc} & 0 & 1 & 2 & \varepsilon \\ \hline q_0 & \{q_0, q_a\} & \{q_1\} & \{q_2\} & \emptyset \\ q_1 & \{q_1\} & \{q_2, q_a\} & \{q_0\} & \emptyset \\ q_2 & \{q_2\} & \{q_0\} & \{q_1, q_a\} & \emptyset \\ q_a & \emptyset & \emptyset & \emptyset & \emptyset \end{array}$$

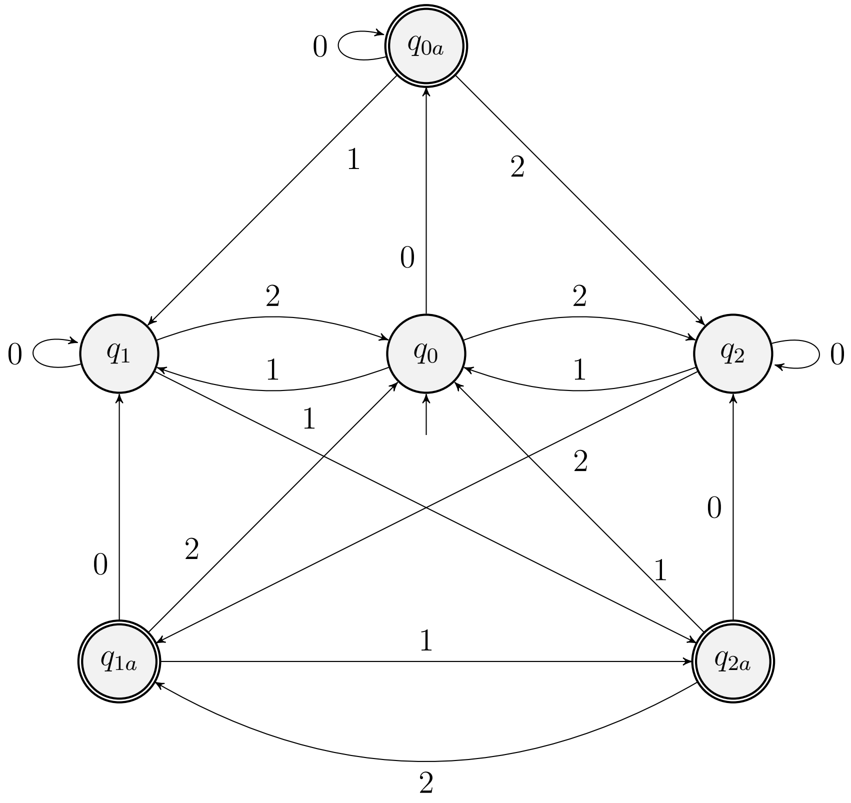
$$F = \{q_a\}$$

The computation of the string “01211” is like the following.



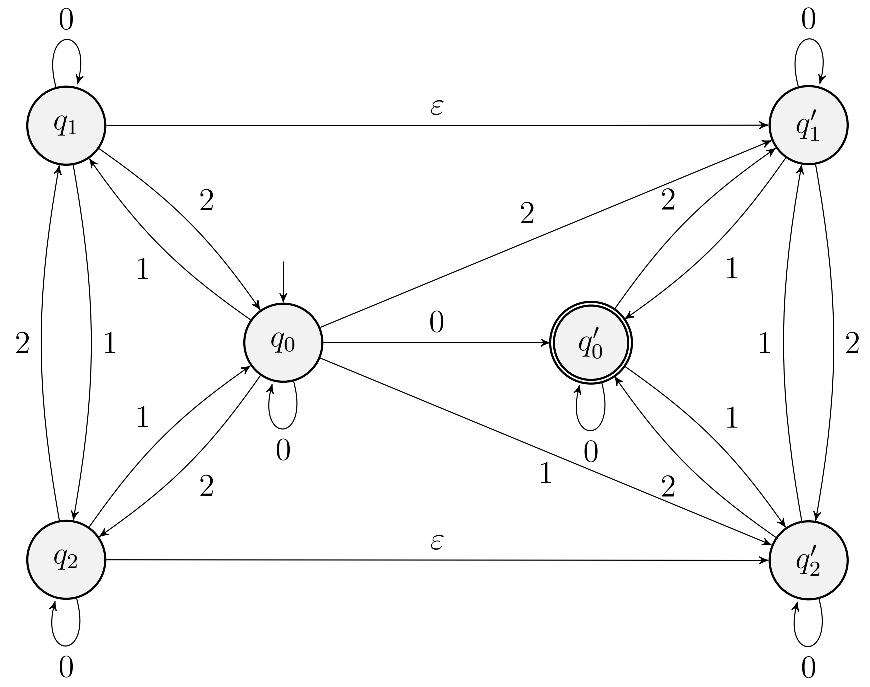
Common mistake: in the formal definition, the ε column is not shown.

(b) See the following diagram

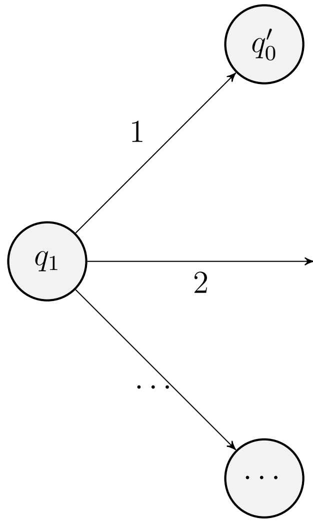


(c) The states q'_1, q'_3, q'_5, q'_7 and q'_9 can be removed because no link goes to these states. This can be seen from the given construction of the δ function: each q'_k is reached via some $\delta(q_i, i)$ or $\delta(q'_i, i)$, and such $k = i + i \mod 10$ must be even.

(d) L_2 can be recognized by the following NFA.



State q_0, q_1 and q_2 are used to add the value of the characters in the first part. Then non-deterministically we go to q_0', q_1' and q_2' and start to subtract the value of characters in the second part. For example, on state q_0' and input character 2, we need a link from q_0' to q_1' since $0 - 2 \mod 3 = 1$. Since $\varepsilon \notin L_2$, we cannot have $q_0 \xrightarrow{\varepsilon} q_0'$ link; instead, we need to specify where to go upon receiving the last character in the first part. Note: Some have links like



This is also fine though an ε link shown above makes the figure simpler.

Problem 3 (20 pts).
LIBSVM[1] is a famous machine learning library. Each line of its input is a given data $(y, \boldsymbol{x})$, where y is the label and $\boldsymbol{x}$ is the feature vector. The following string format represents each $(y, \boldsymbol{x})$:

$$\texttt{<label> <index}_1\texttt{>:<value}_1\texttt{> <index}_2\texttt{>:<value}_2\texttt{>} \cdots \texttt{<index}_n\texttt{>:<value}_n\texttt{>},$$

[1] https://www.csie.ntu.edu.tw/∼cjlin/libsvm/

where `<label>`, `<index>` are integers, and `<value>` is a floating point number. For example, we can utilize

$$\texttt{1 4:1.3 5:2.8} \tag{1}$$

to denote a data

$$\begin{aligned} y &= 1 \\ \boldsymbol{x} &= \begin{bmatrix} 0 & 0 & 0 & 1.3 & 2.8 & 0 & \cdots \end{bmatrix} \end{aligned}$$

Note that zero features are not be stored. Now, we simplify the format with the following alphabet

$$\Sigma = \{\texttt{I}, \texttt{F}, \texttt{:}, \texttt{B}\},$$

where $\texttt{I}, \texttt{F}, \texttt{:}$ and $\texttt{B}$ indicate an integer, a floating point number, a colon and a blank, respectively. Thus, the example (1) becomes

$$\texttt{IBI:FBI:F}.$$

Therefore, the simplified regular expression of the LIBSVM format is

$$\texttt{I(BI:F)}^*. \tag{2}$$

(a) (10 pts) Please generate the NFA that recognizes (2) by the procedure of Fig. 1.57, which is located at

   (i) pages 1-3 of the slide “chap1_regexp2.pdf” and

   (ii) page 68 of the textbook.

   You must complete the diagram step-by-step and show the details. Your resulting diagram should have 11 states.

(b) (10 pts) Let us expand the format to store an instance $(\boldsymbol{y}, \boldsymbol{x})$ that has multiple labels:

$$\texttt{<label}_1\texttt{>,}\cdots\texttt{,<label}_m\texttt{>} \;\; \texttt{<index}_1\texttt{>:<value}_1\texttt{>} \;\; \cdots \;\; \texttt{<index}_n\texttt{>:<value}_n\texttt{>}.$$

   For example,

$$\texttt{1,5,7 4:1.3 5:2.8} \tag{3}$$

   indicates that the instance is associated with labels $1, 5, 7$. Thus, $(\boldsymbol{y}, \boldsymbol{x})$ are

$$\begin{aligned} \boldsymbol{y} &= \begin{bmatrix} 1 & 0 & 0 & 0 & 1 & 0 & 1 & 0 & \cdots \end{bmatrix} \\ \boldsymbol{x} &= \begin{bmatrix} 0 & 0 & 0 & 1.3 & 2.8 & 0 & \cdots \end{bmatrix}. \end{aligned}$$

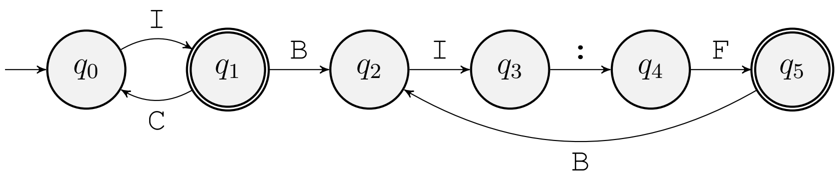
   Similarly, we simplify the format with the alphabets

$$\Sigma \cup \{\texttt{C}\},$$

   where $\texttt{C}$ is denoted as a comma, so that (3) becomes

$$\texttt{ICICIBI:FBI:F}.$$

   Now, TAs have drawn a diagram to recognize this expanded format:



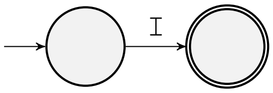
Please help TAs to convert this diagram to regular expression by GNFA. Note that, you need to remove the states with the order

$$q_0, q_1, \ldots, q_5.$$

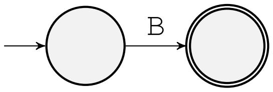
Solution.

(a) Follow the procedure of Fig. 1.57, we construct an NFA with the following steps:

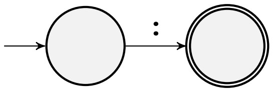
Step 1. I



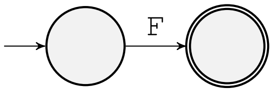
Step 2. B



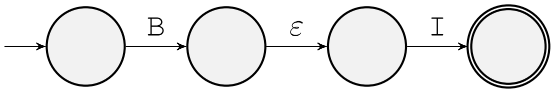
Step 3. :



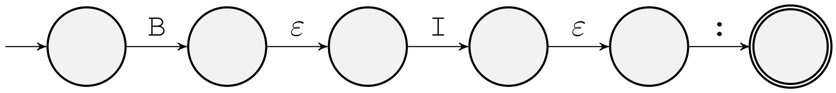
Step 4. F



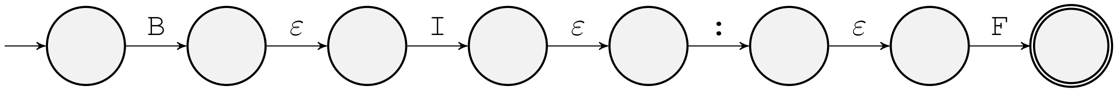
Step 5. BI



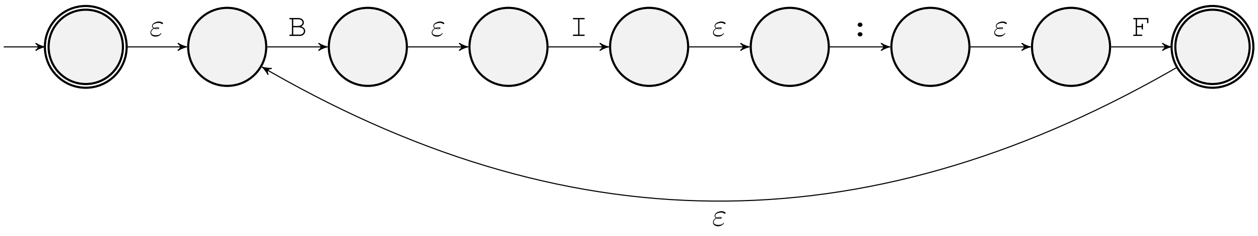
Step 6. BI:



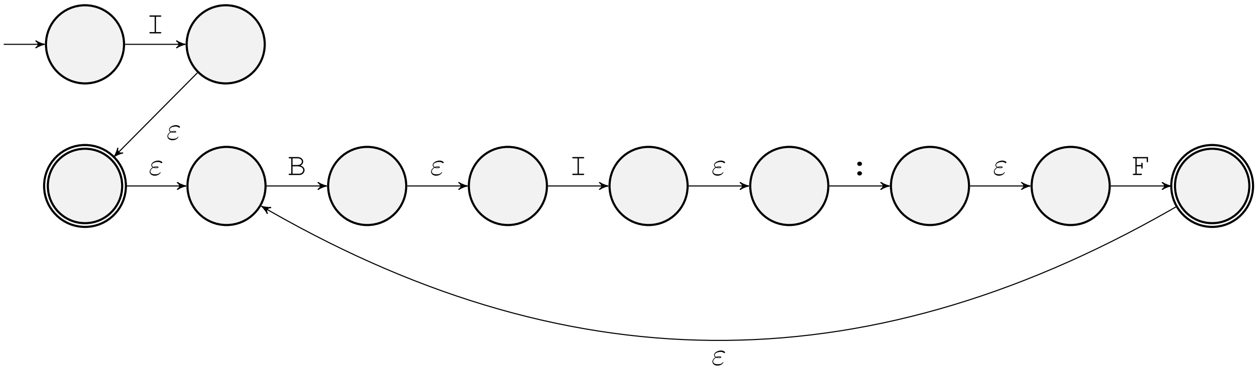
Step 7. BI:F



Step 8. (BI:F)*

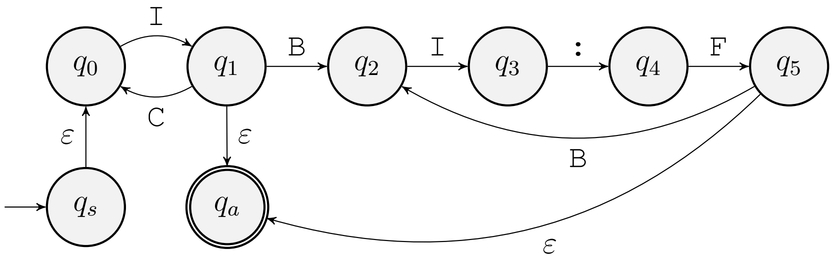


Step 9. $\texttt{I(BI:F)}^*$

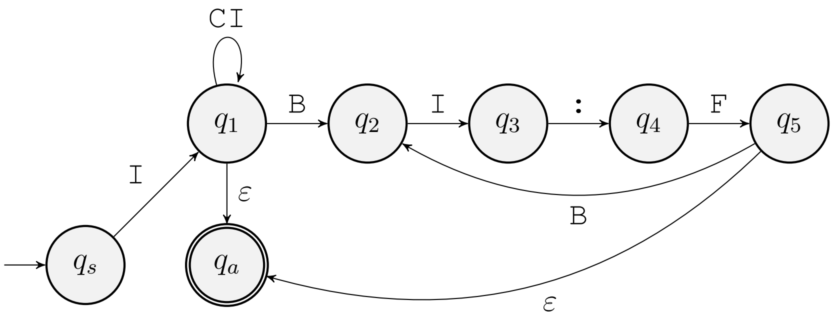


(b) We have the following steps:

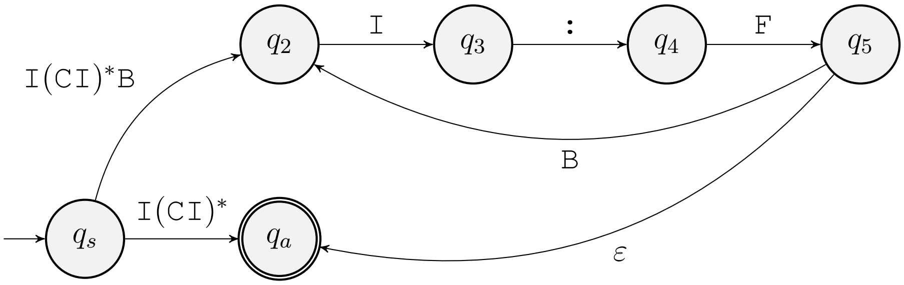
Step 1. Add new start and accept states.



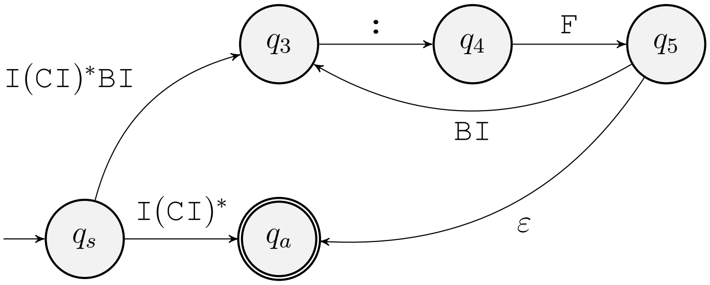
Step 2. Remove q_0.



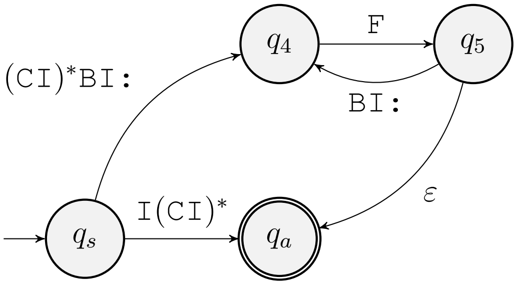
Step 3. Remove q_1.



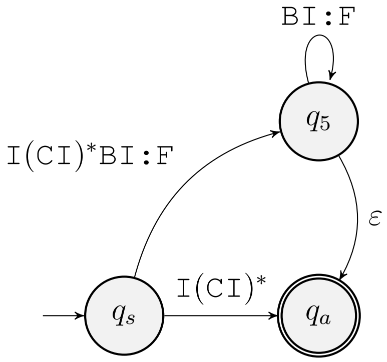
Step 4. Remove q_2.



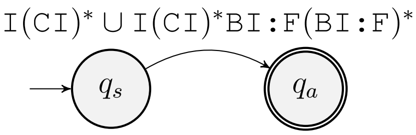
Step 5. Remove q_3.



Step 6. Remove q_4.



Step 7. Remove q_5.



Problem 4 (25 pts). Given $\Sigma = \{0, 1\}$, prove that the following languages are irregular.

(a) (5 pts) The language

$$A = \{0^m 1^n \mid \frac{m}{n} \text{ is a prime }, n \in \mathbb{N}\}.$$

Moreover, let us give

$$s = 0^{2p}1^p$$

with pumping length p, and prove by applying

$$xy^0z \notin A$$

for all partitions x, y and z in s.

(b) (10 pts) In learning the pumping lemma, we know that we need to "guess" an s. Now Xiao-Ming considers

$$s = 0^m 1^p \in A$$

with

$$m = P_1 p, \tag{4}$$

where P_1 is any prime number. For such an s, can we show that s cannot be pumped for all possibilities of x, y and z no matter what P_1 is? In other words, we now show that indeed a set of strings cannot be pumped.

(c) (10 pts) The language

$$B = \left\{0^{m+n}1^n \mid m, n \in \{x_k\}\right\},$$

where the sequence $\{x_k\}$ is defined as

$$\begin{aligned} &x_0 = 0, \\ &x_1 = 1, \\ &x_k = x_{k-1} + x_{k-2}, \ \forall k \geq 2. \end{aligned}$$

Note that you can directly use the fact that $\{x_k\}$ is unbounded and increasing, and here is the proof. By mathematical induction, we can easily prove that

$$x_k > 0, \ \forall k \geq 1.$$

Since

$$x_k = x_{k-1} + x_{k-2},$$

it implies that

$$x_k - x_{k-1} = x_{k-2} > 0, \ \forall k \geq 3. \tag{5}$$

Thus, we have

$$x_k > x_{k-1}, \ \forall k \geq 3,$$

so $\{x_k\}$ is a strictly increasing sequence, which implies the difference between x_k and x_{k-1}

$$x_k - x_{k-1} = x_{k-2}$$

is also strictly increasing as k is growing.

Hint: try to check a few possible s. If chosen properly, the rest of the proof may be quite simple.

Solution.

(a) Assume for contradiction that A is regular with pumping length p. Consider the string

$$s = 0^{2p}1^p \in A.$$

Clearly $|s| \geq p$. By the pumping lemma, s can be written as $s = xyz$ with $|xy| \leq p$, $y > 0$ and $xy^iz \in A$, $\forall i \geq 0$. Because $|xy| \leq p$, y must be of the form

$$0^a, \text{where } 0 < a \leq p.$$

Therefore, xy^0z must be of the form

$$0^{2p-a}1^p, \text{where } 0 < a \leq p.$$

Since $1 \leq \dfrac{2p-a}{p} < 2$, $\dfrac{2p-a}{p}$ must not be a prime and hence $xy^0z \notin A$ for all possible x, y and z.

(b) Suppose A is regular. Given a pumping length p, we consider any prime number P_1 and let

$$m = P_1 p. \tag{6}$$

Then, we have

$$s = 0^m 1^p \in A.$$

Moreover, since the smallest prime number is 2, we can imply that

$$m = P_1 \cdot p \geq 2p. \tag{7}$$

By the lemma, s can be split to

$$s = xyz$$

such that

$$xy^iz \in A,\ \forall i \geq 0, |y| > 0, \text{ and } |xy| \leq p.$$

Because $|xy| \leq p$ and (7), we only need to consider the case $y = 0 \cdots 0$. Therefore,

$$x = 0^a, y = 0^b, z = 0^{m-a-b}1^p,$$

where

$$a \geq 0, m - a \geq b \geq 1, a + b \leq p.$$

Let us check whether

$$xy^iz = 0^{m+(i-1)b}1^p,\ \forall i \geq 0,$$

is in A or not. When

$$i = m + 1 = P_1 p + 1$$

by (6), we have

$$\frac{m + (i-1)b}{p} = \frac{P_1 p + P_1 pb}{p} = P_1(1 + b).$$

Because $b \geq 1$, $P_1(1 + b)$ is not a prime number. Thus, $xy^iz \notin A$ while $i = m + 1$. Therefore, we fail to find xyz with $|y| > 0$ and $|xy| \leq p$ such that

$$xy^iz \in A,\ \forall i \geq 0,$$

so A is not regular.

An alternative solution

Considering the aforementioned notations, and then

$$xy^0z = 0^{P_1p-b}1^p,$$

which implies that

$$\frac{P_1p-b}{p} = P_1 - \frac{b}{p}$$

is a prime number. If

$$0 < |y| = b < p,$$

then

$$P_1 - \frac{b}{p}$$

is not an integer. Thus,

$$xy^0z \notin A.$$

If

$$|y| = p,$$

then

$$P_1 - \frac{b}{p} = P_1 - 1.$$

If $P_1 = 2$, then $P_1 - 1 = 1$ is not a prime number. If $P_1 > 3$, then $P_1 - 1$ is an even integer, which is not a prime number. However, an issue is that when $P_1 = 3$, $P_1 - 1 = 2$ is a prime number. Thus, the case of $P_1 = 3$ must be separately considered. **A common error is that this case is not considered.** From this, let's consider

$$xy^2z = 0^{P_1p+b}1^p,$$

where

$$\frac{P_1p+b}{p} = P_1 + \frac{b}{p}.$$

To have an integer, we still need $|y| = b = p$. For $P_1 \geq 3$, $P_1 + 1$ is an even integer, and is not a prime number. Thus,

$$xy^2z \notin A.$$

For the case of $P_1 = 2$, it has been handled in (a).

Clearly, we still need to have

$$\begin{cases} P_1 = 2 & \text{using } xy^0z \\ P_1 > 2 & \text{using } xy^2z \end{cases}$$

However, for the solution of considering

$$i = P_1p + 1,$$

there is no need to have different cases.

(c) Assume for contradiction that B is regular with pumping length p. Because $\{x_k\}$ is unbounded and increasing, there are elements in $\{x_k\}$ with larger than p. Let p' be the smallest number in $\{x_k\}$ with $p' \geq p$. Consider the string

$$s = 0^{0+p'}1^{p'} \in B.$$

Clearly $|s| \geq p$. By the pumping lemma, s can be written as $s = xyz$ with $|xy| \leq p$, $|y| > 0$ and $xy^iz \in A$, $\forall i \geq 0$. Because $|xy| \leq p$, y must be of the form

$$0^a, \text{where } 0 < a \leq p.$$

Therefore, xy^0z must be of the form

$$0^{p'-a}1^{p'}, \text{where } 0 < a \leq p.$$

However, because the number of 0's must be no less than the number of 1's for each string in B, we have $0^{p'-a}1^{p'} \notin B$ for all possibilities of x, y and z.

An alternative solution

Given a pumping length $p \in \{x_k\}$, we take

$$s = 0^{m+p}1^p = xyz$$

such that

$$xy^iz \in C,\ \forall i \geq 0, |y| > 0, \text{ and } |xy| \leq p.$$

Because $|xy| \leq p$, we only need to consider the case $y = 0 \cdots 0$. Therefore,

$$x = 0^a, y = 0^b, z = 0^{m+p-a-b}1^p,$$

where

$$a \geq 0, b \geq 1, a + b \leq p.$$

Since

$$xy^iz = 0^a0^{ib}0^{m+p-a-b}1^p = 0^{m+p+(i-1)b}1^p \in C,$$

it implies

$$m + (i-1)b \in \{x_k\},\ \forall i \geq 0.$$

We assume these values correspond to

$$x_{k_0}, x_{k_1}, \ldots, x_{k_i}, \ldots$$

in $\{x_k\}$. We have

$$x_{k_{i+1}} - x_{k_i} = b \leq p,\ \forall i \geq 0. \tag{8}$$

Because we showed earlier that $\{x_k\}$ is strictly increasing, there exists an index $\bar{k}$ such that

$$x_k > p, \forall k \geq \bar{k}.$$

Therefore,

$$x_{k+2} - x_{k+1} = x_k > x_{\bar{k}} > p, \forall k \geq \bar{k}. \tag{9}$$

Because $\{k_i \mid \forall i \geq 0\}$ is a strictly increasing index sequence, there exists $\bar{i}$ such that

$$k_j \geq \bar{k},\ \forall j \geq \bar{i}.$$

Then, from (5) and (9)

$$x_{k_{j+1}} - x_{k_j} \geq x_{k_j+1} - x_{k_j} > p,$$

a contradiction to (8). Thus, there exists i such that

$$xy^i z = 0^{m+p+(i-1)b}1^p \notin C.$$

so we fail to find xyz with $|y| > 0$ such that

$$xy^i z \in C, \ \forall i \geq 0.$$

Hence, C is not regular.