Introduction to the Theory of Computation 2025 — Midterm 2

Solutions

Problem 1 (30 pts). Consider the CFG (V, Σ, R, S) with

$$V = \{S, G, H\} \text{ and } \Sigma = \{g, h\},$$

where the rule set R contains the following rules:

$$\begin{aligned} S &\to gG \mid HhGg \\ G &\to hH \mid HGH \mid \varepsilon \\ H &\to HgH \mid \varepsilon \end{aligned} \tag{1}$$

(a) (5 pts) Please provide leftmost derivations for the input strings

$$ghgghg \text{ and } gg$$

by using CFG (1). What you need to give is a sequence of derivations. No need to draw a tree.

(b) (5 pts) Is the string

$$ghg$$

derived ambiguously in CFG (1)? Please provide your reasons for determining ambiguity in CFG (1).

(c) (10 pts) Convert CFG (1) to CNF by the following steps:

(i) Add a new start state.

(ii) Remove $X \to \varepsilon$ with the order

$$G \to \varepsilon,\ H \to \varepsilon,$$

for any variable X that is not the start state.

(iii) Handle $X \to Y$, for all variables X and Y. Please follow the order

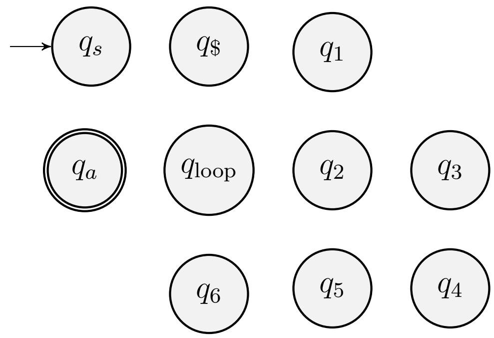
$$G \to G,\ G \to H,\ G \to S,\ H \to G,\ H \to H,\ H \to S,\ S \to G,\ S \to H,\ S \to S,$$
$$S_0 \to G,\ S_0 \to H,\ S_0 \to S.$$

(iv) Convert $X \to u_1 u_2 u_3$, where $k \geq 3$ and each u_i is a variable or terminal symbol.

(v) Replace any terminal u_i in the preceding rules with $U_1 \to g$ and $U_2 \to h$.

For simplicity, you only need to complete steps (i), (ii), and (iii). Steps (iv) and (v) are not required. Please ensure that all intermediate steps are clearly documented.

(d) (10 pts) Please convert CFG (1) to a PDA with the following draft



by the given procedure.

i. Use \$ to ensure that before accepting any string, stack is empty.

ii. Push the start variable S.

iii. Use q_{loop} to handle rules and process input characters.

iv. Replace the left-hand side variable with the right-hand side string for rule substitution.

Please note that we do not allow adding states. With the given procedure, you may find that more states seem necessary. However, some states handle the same rules. Please combine such states so that your construction matches the provided draft.

Solution.

(a) We show the leftmost derivations of those strings on the following.

(i) $ghgghg$.

$$\begin{aligned} S &\rightarrow HhGg \rightarrow HgHhGg \rightarrow \varepsilon gHhGg \rightarrow \varepsilon g\varepsilon hGg \\ &\rightarrow \varepsilon g\varepsilon hHGHg \rightarrow \varepsilon g\varepsilon hHgHGHg \rightarrow \varepsilon g\varepsilon h\varepsilon gHGHg \\ &\rightarrow \varepsilon g\varepsilon h\varepsilon gHgHGHg \rightarrow \varepsilon g\varepsilon h\varepsilon g\varepsilon gHGHg \rightarrow \varepsilon g\varepsilon h\varepsilon g\varepsilon g\varepsilon GHg \\ &\rightarrow \varepsilon g\varepsilon h\varepsilon g\varepsilon g\varepsilon hHHg \rightarrow \varepsilon g\varepsilon h\varepsilon g\varepsilon g\varepsilon h\varepsilon Hg \rightarrow \varepsilon g\varepsilon h\varepsilon g\varepsilon g\varepsilon h\varepsilon\varepsilon g. \end{aligned}$$

(ii) gg.

$$S \rightarrow gG \rightarrow gHGH \rightarrow gHgHGH \rightarrow g\varepsilon gHGH \rightarrow g\varepsilon g\varepsilon GH \rightarrow g\varepsilon g\varepsilon\varepsilon H \rightarrow g\varepsilon g\varepsilon\varepsilon\varepsilon.$$

(b) Since ghg can be derived by the following leftmost derivations

(i)

$$S \rightarrow gG \rightarrow ghH \rightarrow ghHgH \rightarrow gh\varepsilon gH \rightarrow gh\varepsilon g\varepsilon, \text{and}$$

(ii)

$$S \rightarrow HhGg \rightarrow HgHhGg \rightarrow \varepsilon gHhGg \rightarrow \varepsilon g\varepsilon hGg \rightarrow \varepsilon g\varepsilon h\varepsilon g,$$

ghg is derived ambiguously in CFG (1)

(c)
- Add $S_0 \to S$.

$$\begin{aligned} S_0 &\to S \\ S &\to gG \mid HhGg \\ G &\to hH \mid HGH \mid \varepsilon \\ H &\to HgH \mid \varepsilon \end{aligned}$$

- Remove $G \to \varepsilon$

$$\begin{aligned} S_0 &\to S \\ S &\to gG \mid HhGg \mid g \mid Hhg \\ G &\to hH \mid HGH \mid HH \\ H &\to HgH \mid \varepsilon \end{aligned}$$

- Remove $H \to \varepsilon$

$$\begin{aligned} S_0 &\to S \\ S &\to gG \mid HhGg \mid g \mid Hhg \mid hGg \mid hg \\ G &\to hH \mid HGH \mid HH \mid h \mid GH \mid HG \mid G \mid H \\ H &\to HgH \mid gH \mid Hg \mid g \end{aligned}$$

- Remove $G \to G$

$$\begin{aligned} S_0 &\to S \\ S &\to gG \mid HhGg \mid g \mid Hhg \mid hGg \mid hg \\ G &\to hH \mid HGH \mid HH \mid h \mid GH \mid HG \mid H \\ H &\to HgH \mid gH \mid Hg \mid g \end{aligned}$$

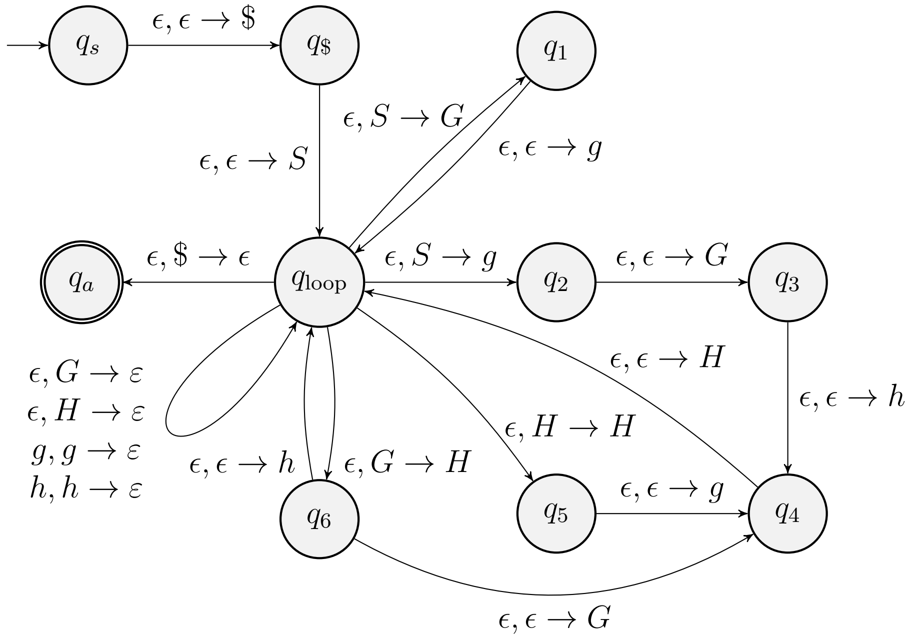
- Remove $G \to H$

$$\begin{aligned} S_0 &\to S \\ S &\to gG \mid HhGg \mid g \mid Hhg \mid hGg \mid hg \\ G &\to hH \mid HGH \mid HH \mid h \mid GH \mid HG \mid HgH \mid gH \mid Hg \mid g \\ H &\to HgH \mid gH \mid Hg \mid g \end{aligned}$$

- Remove $S_0 \to S$

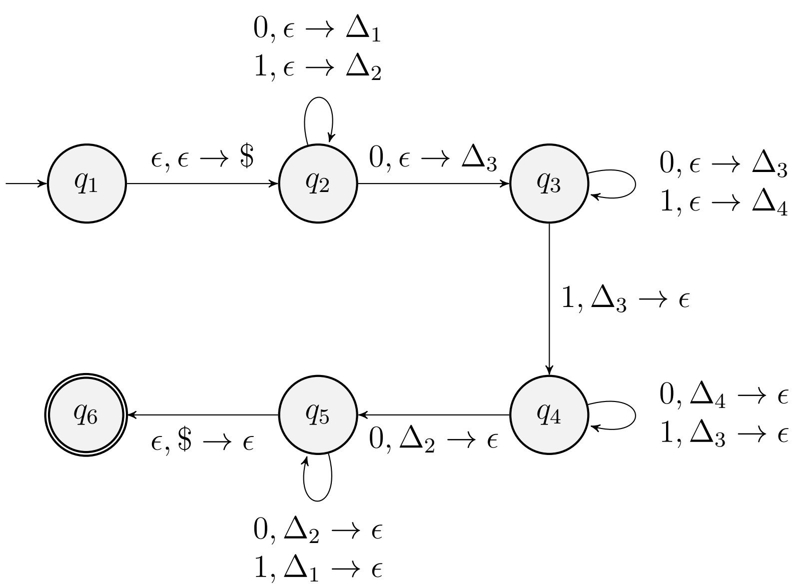
$$\begin{aligned} S_0 &\to gG \mid HhGg \mid g \mid Hhg \mid hGg \mid hg \\ S &\to gG \mid HhGg \mid g \mid Hhg \mid hGg \mid hg \\ G &\to hH \mid HGH \mid HH \mid h \mid GH \mid HG \mid HgH \mid gH \mid Hg \mid g \\ H &\to HgH \mid gH \mid Hg \mid g \end{aligned}$$

(d) Please see the following diagram.



Problem 2 (30 pts). Please consider the following PDA P with

- $Q = \{q_1, q_2, q_3, q_4, q_5, q_6\}$ is the set of states.
- $\Sigma = \{0, 1\}$ is the input alphabet.
- $\Gamma = \{\$, \Delta_1, \Delta_2, \Delta_3, \Delta_4\}$ is the stack alphabet.



(a) (10 pts) Please simulate the given PDA P on the input string **1010** by drawing the corresponding simulation trees. Then, determine whether the PDA P **accepts** this input string based on your simulation.

(b) (10 pts) What is the language recognized by P? Please provide the details to explain your answer. In case you need it, for a string w of length n over $\Sigma = \{0, 1\}$, we define:

- $w^{\mathcal{R}}$ is the string obtained by writing w in the opposite order (i.e., $w_n w_{n-1} \cdots w_1$).
- $\overline{w}$ is the string obtained by flipping (i.e., $1 \to 0, 0 \to 1$) every symbol in w.

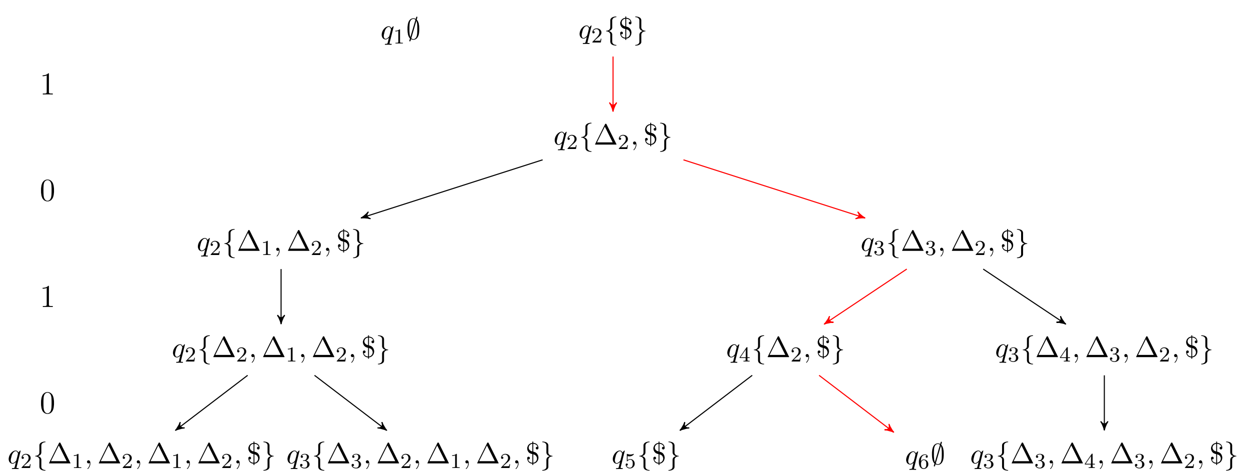
(c) (10 pts) Please convert P to a CFG by utilizing the following method.

> Say that $P = (Q, \Sigma, \Gamma, \delta, q_0, \{q_{accept}\})$ and construct G. The variables of G are $\{A_{pq} \mid p, q \in Q\}$. The start variable is $A_{q_0, q_{accept}}$. Now we describe G's rules in three parts.
>
> (a) For each $p, q, r, s \in Q, u \in \Gamma$, and $a, b \in \Sigma_\varepsilon$, if $\delta(p, a, \varepsilon)$ contains r, u and $\delta(s, b, u)$ contains (q, ε), put the rule $A_{pq} \to aA_{rs}b$ in G.
>
> (b) For each $p, q, r \in Q$, put the rule $A_{pq} \to A_{pr}A_{rq}$ in G.
>
> (c) Finally, for each $p \in Q$, put the rule $A_{pp} \to \varepsilon$ in G.

For simplicity, you only need to write each $A_{pq} \to aA_{rs}b$ rule. The rules $A_{pq} \to A_{pr}A_{rq}$ and $A_{pp} \to \epsilon$ are not needed. In order to prepare for $A_{pq} \to aA_{rs}b$ rules, please give table(s) for each stack alphabet t pushed/popped, similar to what we had in slides.

Solution.

(a) For input string **1010**



This PDA **accepts** input string **1010**.

(b) We observe that states q_2 and q_5 push Δ_1 and Δ_2 into the stack while reading 0s and 1s and then pop Δ_2 and Δ_1 when reading 0s and 1s. This results in the string handled by q_2 being reversed and flipped. However, since state q_4 needs to read a 0 and pop Δ_2 into the stack to reach q_5, the last symbol handled by q_2 must be a 1. As a result, we can summarize that states q_2 and q_5 handle the following language:

$$\{x\overline{x^{\mathcal{R}}} \mid x \in \Sigma^*, x \text{ ends with } 1\}.$$

States q_3 and q_4 perform a similar process as above. They push Δ_3 and Δ_4 into the stack while reading 0s and 1s and then pop Δ_4 and Δ_3 when reading 0s and 1s. This results in the string handled by q_3 being reversed and flipped. Similar to above, since state q_3 needs to read a 1 and pop Δ_4 into the stack to reach q_4, the last symbol handled by q_3 must be a 0. As a result, we can summarize that states q_3 and q_4 handle the following language:

$$\{y\overline{y^{\mathcal{R}}} \mid y \in \Sigma^*, y \text{ ends with } 0\}.$$

We can also notice that state q_2 needs to read a 0 and push Δ_3 into the stack to reach q_3, which needs to be flipped before reaching q_5. Thus, we need a 0 at the beginning of the string handled by q_3. As a result, states q_3 and q_4 with the link $q_2 \to q_3$ are actually handling the following language:

$$\{y\overline{y^{\mathcal{R}}} \mid y \in \Sigma^*, y \text{ starts and ends with } 0\}.$$

Since we handle q_3 and q_4 between q_2 and q_5, the PDA P recognizes the following language:

$$\{x(y\overline{y^{\mathcal{R}}})\overline{x^{\mathcal{R}}} \mid x, y \in \Sigma^*, x \text{ ends with } 1, y \text{ starts and ends with } 0\}.$$

Other solutions:

- $$\{x1y\overline{y^{\mathcal{R}}}0\overline{x^{\mathcal{R}}} \mid x, y \in \Sigma^*, y \text{ starts and ends with } 0\}.$$
- $$\{xy\overline{(xy)^{\mathcal{R}}} \mid x, y \in \Sigma^*, x \text{ ends with } 1, y \text{ starts and ends with } 0\}.$$

(c) The corresponding CFG is the following:

- $t = \Delta_1$:

p	r	s	q	a	b	rules
2	2	5	5	0	1	$A_{25} \to 0A_{25}1$

- $t = \Delta_2$:

p	r	s	q	a	b	rules
2	2	4	5	1	0	$A_{25} \to 1A_{24}0$
2	2	5	5	1	0	$A_{25} \to 1A_{25}0$

- $t = \Delta_3$:

p	r	s	q	a	b	rules
2	3	3	4	0	1	$A_{24} \to 0A_{33}1$
2	3	4	4	0	1	$A_{24} \to 0A_{34}1$
3	3	3	4	0	1	$A_{34} \to 0A_{33}1$
3	3	4	4	0	1	$A_{34} \to 0A_{34}1$

- $t = \Delta_4$:

p	r	s	q	a	b	rules
3	3	4	4	1	0	$A_{34} \to 1A_{34}0$

- $t = \$$:

p	r	s	q	a	b	rules
1	2	5	6	ϵ	ϵ	$A_{16} \to A_{25}$

Problem 3 (20 pts). We have a special programming language that uses the notations

- △ for changing the line into the next one,
- We use two "$$$" for giving some comments in the program, e.g.,

  I-am-not-comments.$$$These-are△
  △
  comments-with-multiple-lines△
  $$$△ (2)
  I-am-not-comments.△
  We-have-empty-comment-on-the-right.$$$$$$△

  contains the comments between two "$$$" signs.

Moreover, the comments should follow the rules below:

- In the end of the comments, the notation $$$ must be followed by the notation △.
- $ cannot be used in the comment content.
- The notation △ can only represent for changing the line.

In this problem, you need to design a PDA to check whether the usage of the comments is valid. Since the PDAs can only input a one-line string, we can treat (2) as

I-am-not-comments.$$$These-are △ △comments-with-multiple-lines △ $$$ △ I-am-not-comments. △We-have-empty-comment-on-the-right.$$$$$$△

to the PDAs.

Now, let us check more examples.

- A **valid** program

  Automata-is-my-favorite-course.△
  $$$Midterm-1-is-so-easy.$$$△

  satisfy all the rules.

- Another **valid** program

  Automata-is-my-favorite-course.△

  can be accepted since we do not have comments in the program.

- Another **valid** program

  Automata-is-my-favorite-course.

  can be accepted since we do not have comments in the program.

- An **invalid** program

  Automata-is-my-favorite-course.△
  $$$△
  I-am-wrong-with△
  incomplete-notations.△
  $$△

  lacks of a $ in the second $$$ sign.

- Another **invalid** program

  Automata-is-my-favorite-course.△
  $$$△
  $-cannot-be-used$△
  in-the-comments.△
  $$$△

  uses $ in the comment content, which violates the rule

  "$ cannot be used in the comment content."

- Another **invalid** program

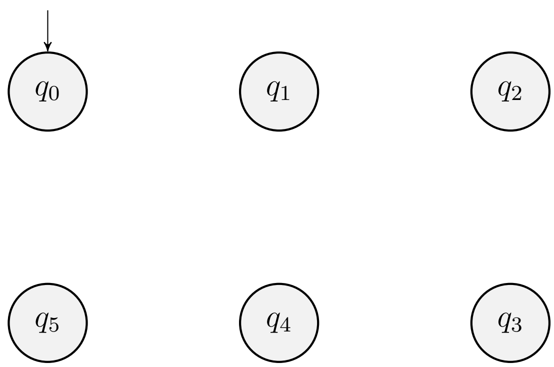
Automata-is-my-favorite-course.△
$$$TAs-are-friendly.$$$This-place-cannot-be-writen△

violates the rule

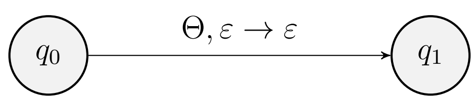
"In the end of the comments, the notation $$$ must be followed by the notation △."

(a) (15 pts) Please finish your PDA with

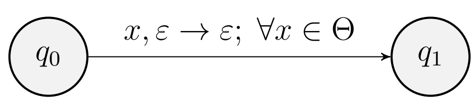
- Θ denotes the set of all the used characters in the program, but $\$ \notin \Theta$. Note that $\triangle \in \Theta$.
- $\Sigma = \Theta \cup \{\$\}$,
- $\Gamma = \{\$, \triangle\}$, and
- The draft



You are allowed to use the representation



to stand for



Your diagram must have no more than six states. We found ways to use either five or six states.

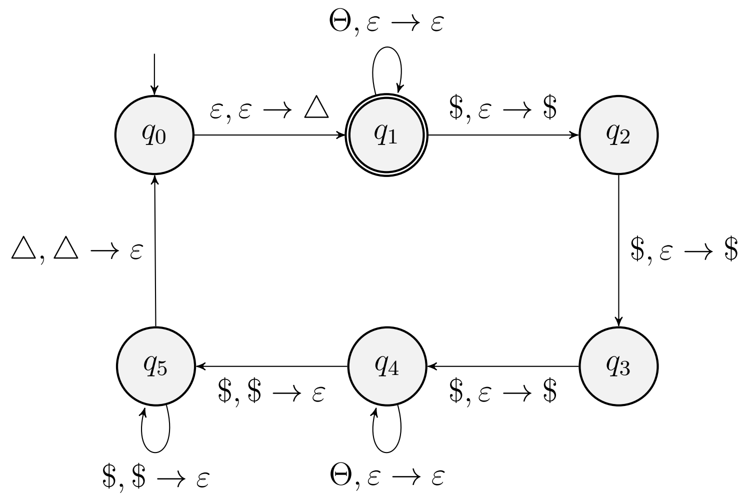
(b) (5 pts) Please check whether your PDA in (a) can accept the string

$$a\$\$\$b\$\$\$\triangle$$

To simplify the problem, you only need to write down the path that leads to the acceptance.

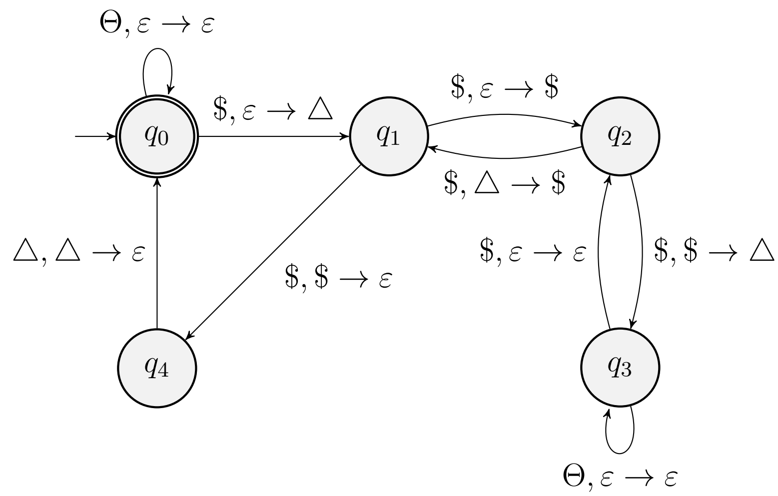
Solution.

(a) In the started node q_0, we append the character △ into the stack to identify whether we are in the beginning of the stack. In node q_1, we process the content of the program. The nodes q_2 and q_3 handle the previous comment notations and append $s into the stack. The node q_4 processes the content of the comments. The node q_5 make sure that we pop out three $s and go back to the node q_0 to continuously load the program.

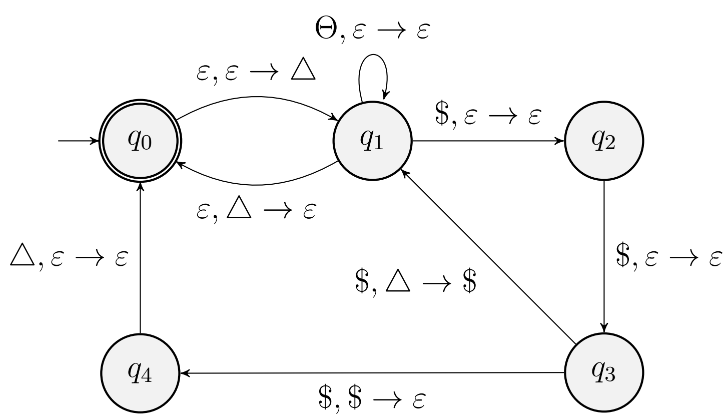


Other solutions:

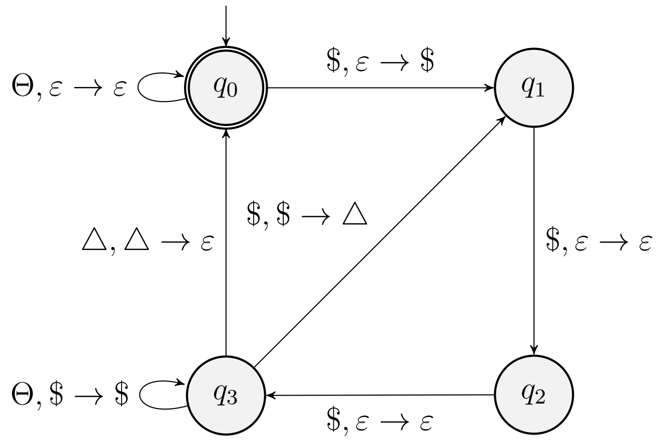
- Other solution 1.



- Other solution 2.



- Actually, we can use a 4-node PDA to solve this problem.



(b) Here is the simulation:

$$\to q_0\ \{\} \xrightarrow{\varepsilon} q_1\{\triangle\} \xrightarrow{a} q_1\{\triangle\} \xrightarrow{\$} q_2\{\triangle\$\} \xrightarrow{\$} q_3\{\triangle\$\$\} \xrightarrow{\$} q_4\{\triangle\$\$\$\} \xrightarrow{b} q_4\{\triangle\$\$\$\}$$
$$\xrightarrow{\$} q_5\{\triangle\$\$\} \xrightarrow{\$} q_5\{\triangle\$\} \xrightarrow{\$} q_5\{\triangle\} \xrightarrow{\triangle} q_0\{\} \xrightarrow{\varepsilon} q_1\{\triangle\} \to \text{accepted}$$

Problem 4 (20 pts). In this problem, you will design a TM for the task "string masking." For example, in the beginning of the TM, we have the tape

$$a \quad b \quad a \quad a \quad b \quad \# \quad 0 \quad 1 \quad 0 \quad 1 \quad 0 \quad \# \quad \sqcup \quad \cdots$$

which represents we have the input string

$$abaab$$

and the mask

$$01010.$$

After the masking, i.e.,

$$\begin{matrix} a & b & a & a & b \\ 0 & 1 & 0 & 1 & 0 \end{matrix} \Rightarrow ba$$

we should add the result in the end of the tape as

$$\cdots \quad \# \quad \cdots \quad \# \quad b \quad a \quad \sqcup \quad \cdots$$

Note that we assume the input tape must be

$$\{a, b\}^*\#\{0, 1\}^*\#$$

Moreover, the input and the mask strings already have the **same** length (i.e., no need to check this).

(a) (15 pts) Please follow the steps to design your TM:

Step 1: Mark the first un-processed character to $\sqcup$, and go to the corresponding mask bit.

Step 2: If the bit is 1, do the following things.

- Modify the bit to $\sqcup$
- Go to the answer part (i.e., the part after the 2nd #) and replace the 1st $\sqcup$ with the target character.

Step 3: Go back to find the next un-processed character and go to Step 1.

Step 4: If all the characters are processed, go to the accepted node.

Note that your TM should satisfy

- $\Sigma = \{\#, a, b, 0, 1, \sqcup\}$,
- $\Gamma = \Sigma$,
- no more than 11 states (the rejected states are excluded), and
- we only consider moving the head right or left in the Turing machine.

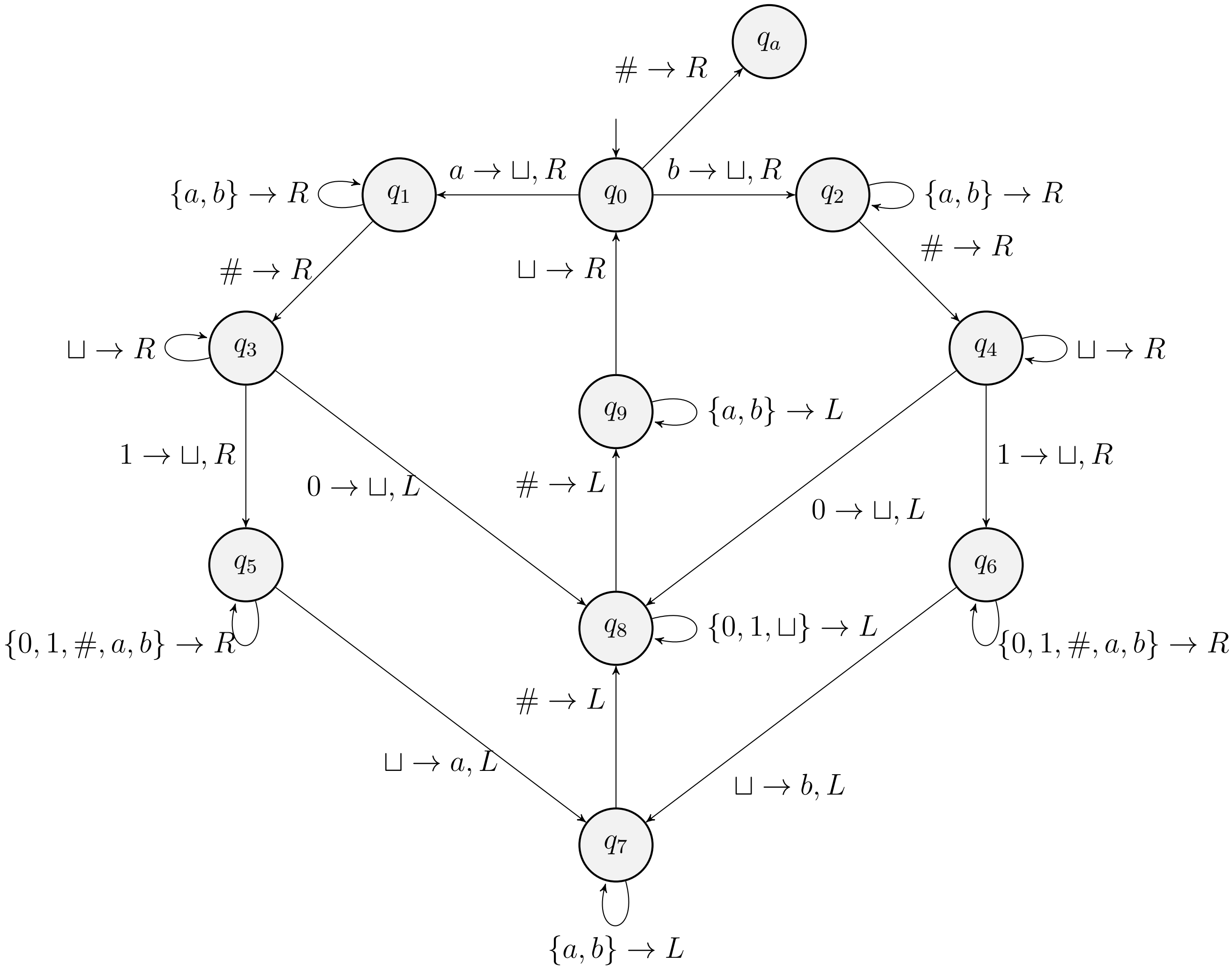
(b) (5 pts) Please simulate your TM in (a) on the string

$$ab\#01\#$$

Note that you need to show the entire simulation until your Turing Machine stops.

Solution.

(a) Since this problem is symetric for masking a and b, we give the explanation on masking a without loss of generality. In q_0, we handle the target character, and pass all other character before the 1st # in the q_1. Then, we pass the processed mask bits in q_3. In the case "the next un-processed mask bit is 1," we utilize q_5 and q_7 for adding the corresponding target character (i.e., a) in the answer part (the part after the 2nd #). Then, we go back to the next target character throught q_7, q_8, and q_9. In another the case "the next un-processed mask bit is 0," we go back to the next target character throught q_8 and q_9. After processing all the characters, we will reach the 1st # in q_0. Thus, we can go to the accepted node.



(b) Here is the simulation:

$q_0ab\#01\# \to \sqcup q_1 b\#01\# \to \sqcup b q_1\#01\# \to \sqcup b\# q_3 01\# \to \sqcup b q_8 \# \sqcup 1\# \to \sqcup q_9 b\# \sqcup 1\# \to q_9 \sqcup b\# \sqcup 1\#$

$\to \sqcup q_0 b\# \sqcup 1\# \to \sqcup \sqcup q_2\# \sqcup 1\# \to \sqcup \sqcup \# q_4 \sqcup 1\# \to \sqcup \sqcup \# \sqcup q_4 1\# \to \sqcup \sqcup \# \sqcup \sqcup q_6\#$

$\to \sqcup \sqcup \# \sqcup \sqcup \# q_6 \sqcup \to \sqcup \sqcup \# \sqcup \sqcup q_7\#b \to \sqcup \sqcup \# \sqcup q_8 \sqcup \#b \to \sqcup \sqcup \# q_8 \sqcup \sqcup \#b$

$\to \sqcup \sqcup q_8\# \sqcup \sqcup \#b \to \sqcup q_9 \sqcup \# \sqcup \sqcup \#b \to \sqcup \sqcup q_0\# \sqcup \sqcup \#b \to \sqcup \sqcup \# q_a \sqcup \sqcup \#b \to$ accepted