Introduction to the Theory of Computation 2024 — Final

Solutions

Problem 1 (10 pts). In Chapter 5 of our textbook, we use the technique of the reducibility to prove

$$\mathrm{HALT}_{\mathrm{TM}} = \{\langle M, \boldsymbol{w}\rangle \mid\ M \text{ is a TM, and } M \text{ halts on the input } \boldsymbol{w}\} \text{ is undecidable.}$$

Here is the proof: Suppose that $\mathrm{HALT}_{\mathrm{TM}}$ is decidable, so we have a decider R that decides $\mathrm{HALT}_{\mathrm{TM}}$. Therefore, we construct a TM S for $A_{\mathrm{TM}} = \{\langle M, \boldsymbol{w}\rangle \mid M \text{ is a TM and } M \text{ accepts } \boldsymbol{w}\}$ on the following.

- On input $\langle M, \boldsymbol{w}\rangle$ (an encoding of a TM M and a string $\boldsymbol{w}$):
  - 1° Run TM R on input $\langle M, \boldsymbol{w}\rangle$.
  - 2° If R rejects $\langle M, \boldsymbol{w}\rangle$, S rejects.
  - 3° If R accepts $\langle M, \boldsymbol{w}\rangle$, S can then simulate M on $\boldsymbol{w}$ until M returns rejection or acceptance.
  - 4° If M accepts $\boldsymbol{w}$, S accepts. Otherwise, S rejects.

If R is a decider, it implies that S decides A_{TM}, which is a contradiction. Hence, $\mathrm{HALT}_{\mathrm{TM}}$ is undecidable.

Now, let us consider another language

$$L_1 = \{\langle M, \boldsymbol{w}\rangle \mid\ M \text{ is a TM, and } M \text{ does not halt on the input } \boldsymbol{w}\}.$$

Please consider the above proof, and use a similar idea to prove that

$$L_1 \text{ is undecidable.}$$

Solution.

Suppose that L_1 is decidable. Then, we have a decider R that decides L_1. Therefore, we construct a TM S for

$$A_{\mathrm{TM}} = \{\langle M, \boldsymbol{w}\rangle \mid M \text{ is a TM and } M \text{ accepts } \boldsymbol{w}\}$$

on the following.

- On input $\langle M, \boldsymbol{w}\rangle$ (an encoding of a TM M and a string $\boldsymbol{w}$):
  - 1° Run TM R on input $\langle M, \boldsymbol{w}\rangle$.
  - 2° If R accepts $\langle M, \boldsymbol{w}\rangle$, S rejects.
  - 3° If R rejects $\langle M, \boldsymbol{w}\rangle$, S can then simulate M on $\boldsymbol{w}$ until M returns rejection or acceptance.
  - 4° If M accepts $\boldsymbol{w}$, S accepts. Otherwise, S rejects.

If R is a decider, it implies that S decides A_{TM}, which is a contradiction. Hence, L_1 is undecidable.

An alternative solution

Suppose that L_1 is decidable. Then, we have a decider R that decides L_1. Therefore, we construct a TM S for HALT_{TM} on the following.

- On input $\langle M, \boldsymbol{w} \rangle$ (an encoding of a TM M and a string $\boldsymbol{w}$):

  1° Run TM R on input $\langle M, \boldsymbol{w} \rangle$.

  2° If R accepts $\langle M, \boldsymbol{w} \rangle$, S rejects.

  3° If R rejects $\langle M, \boldsymbol{w} \rangle$, S accepts.

If R is a decider, it implies that S decides HALT_{TM}, which contradicts our proof that HALT_{TM} is undecidable. Hence, L_1 is undecidable.

Problem 2 (20 pts). Assume f and g are functions $f, g : \mathbb{N} \to \mathbb{R}^+$. Prove or disprove the subproblems by using the following definitions.

- We say $f(n) = O(g(n))$ if there exists $c > 0$ and $n_0 \in \mathbb{N}$ such that for every integer $n \geq n_0$,

$$f(n) \leq cg(n).$$

- We say $f(n) = o(g(n))$ if for each $c > 0$, there exists $n_0 \in \mathbb{N}$ such that for every integer $n \geq n_0$,

$$f(n) \leq cg(n).$$

For proving the statements, you must give the specific n_0 for one c or all c's, depending on the definition of big-O or small-o. For disproving the statements, you must prove the opposite of the definition by also showing details. Note that we use natural log in this problem, i.e., $\log n = \log_e n$.

(a) (5 pts) Let $f(n) = \sqrt{n} \log n$ and $g(n) = n$. Whether $f(n) = O(g(n))$? Hint: You can directly use the following property without any proof.

$$\log n \leq \sqrt{n}, \text{ for all } n = 1, 2, \ldots.$$

(b) (5 pts) Let $f(n) = n!$ and

$$g(n) = e^{n \log n}.$$

Whether $f(n) = O(g(n))$?

(c) (10 pts) Let $f(n) = \log n$ and

$$g(n) = \frac{n}{\log n}.$$

Whether $f(n) = o(g(n))$? Hint: You can directly use the following property without any proof.

$$\log n \leq \sqrt[3]{n}, \text{ for all } n \geq 100.$$

Solution.

(a) We can take $c = 1$ and $n_0 = 1$ such that

$$\log n \leq \sqrt{n} \Rightarrow \sqrt{n}\log n \leq 1\sqrt{n}\sqrt{n} = n$$

for all $n \geq n_0$.

(b) Since

$$e^{n\log n} = e^{\log n^n} = n^n,$$

we can take $c = 1$ and $n_0 = 1$ such that

$$n! = n(n-1)\cdots 1 \leq \underbrace{n\cdots n}_{n} = n^n = 1e^{n\log n},$$

for any $n \geq n_0$.

(c) Since the formulation

$$\log n \leq c\frac{n}{\log n} \equiv (\log n)^2 \leq cn \equiv \left(\frac{\log n}{\sqrt[3]{n}}\right)^2 \leq c\sqrt[3]{n},$$

we prove that

$$c\sqrt[3]{n} - \left(\frac{\log n}{\sqrt[3]{n}}\right)^2 \geq 0$$

on the following.

Given $c > 0$, we can take

$$n_0 = \left\lceil \max\left(\frac{1}{c^3}, 100\right)\right\rceil,$$

which implies

$$\frac{\log n}{\sqrt[3]{n}} \leq 1,\ \forall n \geq n_0 \geq 100$$

from hint and

$$c\sqrt[3]{n} \geq 1,\ \forall n \geq n_0 \geq \frac{1}{c^3}$$

such that

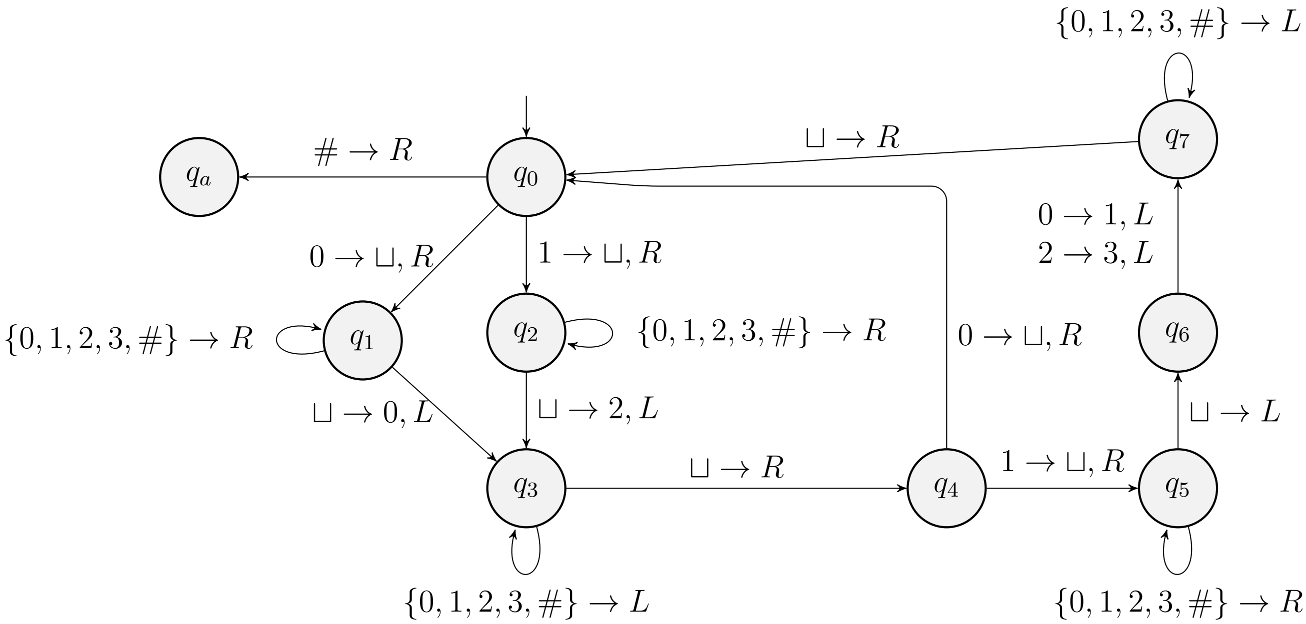
$$c\sqrt[3]{n} - \left(\frac{\log n}{\sqrt[3]{n}}\right)^2 \geq 1 - 1 = 0.$$

Problem 3 (25 pts). In the midterm 2, we have learned how to

mapping a **non-negative** binary number $\boldsymbol{x}$ into an equivalent quaternary number $\boldsymbol{y}$

with the following TM, where

$$\boldsymbol{x} = x_1x_2\cdots x_{2n-1}x_{2n} \in \{0,1\}^{2n} \text{ and } \boldsymbol{y} = y_1y_2\cdots y_n \in \{0,1,2,3\}^n,\ \forall n = 0,1,2,\ldots.$$



Now, you will derive the time complexity of this TM by the following sub-problems. Note that if $x_{2k} = 0$ for some k, the TM will go to q_0 from q_4 by one step, which is less then the steps from

$$q_4 \to q_5 \to q_6 \to q_7 \to q_0.$$

However, we would like to know the maximum number of steps for all $\boldsymbol{x}$, so we only consider the case $x_{2k} = 1$, for all $k = 1, 2, \ldots, n$.

(a) (8 pts) Please derive the steps that the TM uses on the input

$$1111\# \sqcup \cdots$$

by completing the parts (I) $\cdots$ (VIII) of the following table. Note that the 1st round and 2nd round mean to handle the underline bits of

$$\underline{11}11\# \sqcup \text{ and } \sqcup\sqcup\underline{11}\#3\sqcup, \text{ respectively.}$$

terms	1st round	# **step(s)**	2nd round	# **step(s)**
$q_0 \to q_1$ or $q_0 \to q_2$	$\sqcup q_2 111\#\sqcup$	1	$\sqcup\sqcup\sqcup q_2 1\#3\sqcup$	1
$q_1 \to q_1$ or $q_2 \to q_2$	$\sqcup 111\# q_2 \sqcup$	(I)	$\sqcup\sqcup\sqcup 1\#3 q_2 \sqcup$	(V)
$q_1 \to q_3$ or $q_2 \to q_3$	$\sqcup 111 q_3 \#2\sqcup$	1	$\sqcup\sqcup\sqcup 1\# q_3 32\sqcup$	1
$q_3 \to q_3$	$q_3 \sqcup 111\#2\sqcup$	(II)	$\sqcup\sqcup q_3 \sqcup 1\#32\sqcup$	(VI)
$q_3 \to q_4$	$\sqcup q_4 111\#2\sqcup$	1	$\sqcup\sqcup\sqcup q_4 1\#32\sqcup$	1
$q_4 \to q_5$	$\sqcup\sqcup q_5 11\#2\sqcup$	1	$\sqcup\sqcup\sqcup\sqcup q_5 \#32\sqcup$	1
$q_5 \to q_5$	$\sqcup\sqcup 11\#2 q_5 \sqcup$	(III)	$\sqcup\sqcup\sqcup\sqcup \#32 q_5 \sqcup$	(VII)
$q_5 \to q_6$	$\sqcup\sqcup 11\# q_6 2\sqcup$	1	$\sqcup\sqcup\sqcup\sqcup \#3 q_6 2\sqcup$	1
$q_6 \to q_7$	$\sqcup\sqcup 11 q_7 \#3\sqcup$	1	$\sqcup\sqcup\sqcup\sqcup \# q_7 33\sqcup$	1
$q_7 \to q_7$	$\sqcup q_7 \sqcup 11\#3\sqcup$	(IV)	$\sqcup\sqcup\sqcup q_7 \sqcup \#33\sqcup$	(VIII)
$q_7 \to q_0$	$\sqcup\sqcup q_0 11\#3\sqcup$	1	$\sqcup\sqcup\sqcup\sqcup q_0 \#33\sqcup$	1
$q_0 \to q_a$		0	$\sqcup\sqcup\sqcup\sqcup \# q_a 33\sqcup$	1

You can directly write your answers of (I) $\cdots$ (VIII) without the explanation.

(b) (12 pts) Now, we consider the tth round in the general term, and let $x_{2k} = 1$, for all $k = 1, \dots, n$. That is, the tape of the TM is

$$\underbrace{\sqcup \cdots \sqcup}_{2t-2} x_{2t-1} x_{2t} \cdots x_{2n-1} x_{2n} \# y_1 y_2 \cdots y_{t-1} \sqcup \cdots$$

in the beginning. When we finish the tth round, the tape becomes

$$\underbrace{\sqcup \cdots \sqcup}_{2t-2} \sqcup \sqcup x_{2t+1} x_{2t+2} \cdots x_{2n-1} x_{2n} \# y_1 y_2 \cdots y_{t-1} y_t \sqcup \cdots .$$

Please complete the parts (I) $\cdots$ (IV) of the following table.

terms	# **step(s) of the tth round**
$q_0 \to q_1$ or $q_0 \to q_2$:	1
$q_1 \to q_1$ or $q_2 \to q_2$:	(I)
$q_1 \to q_3$ or $q_2 \to q_3$:	1
$q_3 \to q_3$:	(II)
$q_3 \to q_4$:	1
$q_4 \to q_5$:	1
$q_5 \to q_5$:	(III)
$q_5 \to q_6$:	1
$q_6 \to q_7$:	1
$q_7 \to q_7$:	(IV)
$q_7 \to q_0$:	1
$q_0 \to q_a$:	1 for the final round otherwise 0

You **must explain** how to get your answers of (I) $\cdots$ (IV).

(c) (5 pts) Please derive the total number of the steps from 1st round to nth round via the formulation in (b), which means that the total steps of the TM use on the input

$$x_1 x_2 \cdots x_{2n-1} x_{2n} \# \sqcup \cdots ,$$

where $x_{2k-1} \in \{0, 1\}$ and $x_{2k} = 1$ for all $k = 1, \dots, n$.

Solution.

(a) After directly counting from the tape, we know that

$$\text{(I)} = 4,\ \text{(II)} = 4,\ \text{(III)} = 4,\ \text{(IV)} = 3,\ \text{(V)} = 3,\ \text{(VI)} = 3,\ \text{(VII)} = 3,\ \text{(VIII)} = 2.$$

(b) In the tth round, the tape status is

$$\sqcup \cdots \sqcup\, q_0 x_{2t-1} x_{2t} \cdots x_{2n-1} x_{2n} \# y_1 \cdots y_{t-1} \sqcup .$$

Hence, the loop $q_1 \to q_1$ or $q_2 \to q_2$ moves $2n - 2t + 1$ steps from

$$\sqcup \cdots \sqcup \sqcup q_1 x_{2t} \cdots x_{2n-1} x_{2n} \# y_1 \cdots y_{t-1} \sqcup \Rightarrow \sqcup \cdots \sqcup \sqcup x_{2t} \cdots x_{2n-1} x_{2n} q_1 \# y_1 \cdots y_{t-1} \sqcup,$$

and then uses t steps from

$$\sqcup \cdots \sqcup \sqcup x_{2t} \cdots x_{2n-1} x_{2n} q_1 \# y_1 \cdots y_{t-1} \sqcup \Rightarrow \sqcup \cdots \sqcup \sqcup x_{2t} \cdots x_{2n-1} x_{2n} \# y_1 \cdots y_{t-1} q_1 \sqcup .$$

Thus, the total cost is $2n-t+1$ steps in this loop. For the loop $q_3 \to q_3$, we spend $t-1$ steps from

$$\sqcup \cdots \sqcup \sqcup x_{2t} \cdots x_{2n-1}x_{2n}\#y_1 \cdots q_3 y_{t-1}y_t \sqcup \Rightarrow \sqcup \cdots \sqcup \sqcup x_{2t} \cdots x_{2n-1}x_{2n}q_3\#y_1 \cdots y_{t-1}y_t\sqcup,$$

and then using $2n-2t+2$ steps go to

$$\sqcup \cdots \sqcup q_3 \sqcup x_{2t} \cdots x_{2n-1}x_{2n}\#y_1 \cdots y_{t-1}y_t \sqcup .$$

The total cost is $2n-t+1$ in this loop. In the loop $q_5 \to q_5$, it takes $2n-2t$ steps from

$$\sqcup \cdots \sqcup \sqcup \sqcup q_5 x_{2t+1} \cdots x_{2n-1}x_{2n}\#y_1 \cdots y_{t-1}y_t \sqcup \Rightarrow \sqcup \cdots \sqcup \sqcup \sqcup x_{2t+1} \cdots x_{2n-1}x_{2n}q_5\#y_1 \cdots y_{t-1}y_t\sqcup,$$

and further spends $t+1$ steps to

$$\sqcup \cdots \sqcup \sqcup \sqcup x_{2t+1} \cdots x_{2n-1}x_{2n}\#y_1 \cdots y_{t-1}y_t q_5 \sqcup .$$

In the final, the loop $q_7 \to q_7$ spends $t-1$ steps from

$$\sqcup \cdots \sqcup \sqcup \sqcup x_{2t+1} \cdots x_{2n-1}x_{2n}\#y_1 \cdots q_7 y_{t-1}y_t \sqcup \Rightarrow \sqcup \cdots \sqcup \sqcup \sqcup x_{2t+1} \cdots x_{2n-1}x_{2n}q_7\#y_1 \cdots y_{t-1}y_t\sqcup,$$

and uses $2n-2t+1$ to

$$\sqcup \cdots \sqcup \sqcup q_7 \sqcup x_{2t+1} \cdots x_{2n-1}x_{2n}\#y_1 \cdots y_{t-1}y_t \sqcup .$$

The total cost is $2n-t$ in this loop.

Overall, we have

$$\text{(I)} = 2n-t+1,\ \text{(II)} = 2n-t+1,\ \text{(III)} = 2n-t+1,\ \text{(IV)} = 2n-t.$$

(c) In (b), we know that the tth round's cost is

$$8n-4t+10.$$

Since we have n rounds,

$$\sum_{t=1}^{n}(8n-4t+10) = 8n^2 - 4\frac{(1+n)(n)}{2} + 10n = 6n^2+8n.$$

We also contain 1 step for $q_0 \to q_a$, so the total steps is

$$6n^2+8n+1.$$

Problem 4 (20 pts). Deoxyribonucleic Acid (DNA) is consisted with four components

$$\{\text{A}, \text{C}, \text{G}, \text{T}\},$$

and many features of a human may come from a short sequence of his DNA sequence. For example, a sequence of ATTTTG might instruct for blue eyes, while a sequence of TTTTTG might instruct for brown. Therefore, finding a specific sequence from the main sequence becomes an important issue. Now, we hope that you can design a 2-tape NTM to achieve that. Note that we do not teach multi-tape NTM in our course. However, the difference between 2-tape TM and 2-tape NTM is the δ function

$$\begin{array}{ccc} \text{2-tape TM} & \Leftrightarrow & \text{2-tape NTM} \\ \delta : Q \times \Gamma^2 \to Q \times \Gamma^2 \times \{L,R,S\}^2 & \Leftrightarrow & \delta : Q \times \Gamma^2 \to \mathcal{P}(Q \times \Gamma^2 \times \{L,R,S\}^2) \end{array}.$$

Let us consider $\Sigma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \#\}$, $\Gamma = \{\mathrm{A}, \mathrm{C}, \mathrm{G}, \mathrm{T}, \#, \sqcup\}$, and stop, i.e., we have $\{L, R, S\}$. Note that if the head points at the first position and moves left, it will **still be in the same location.** For the inputs, we have

DNA_sequence#target_sequence

in the 1st tape, where both DNA_sequence and target_sequence are **not empty**. For example,

AAAAAAAAAA#AC.

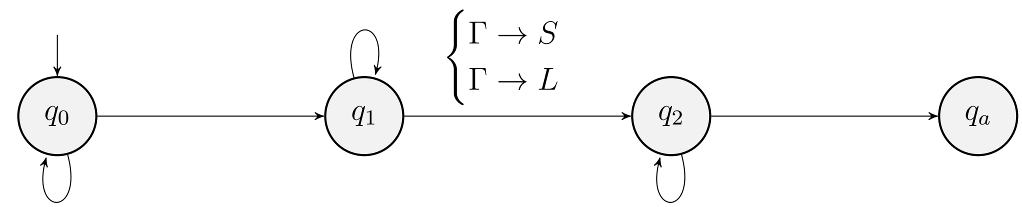
(a) (10 pts) Please follow the procedure

1° Copy DNA_sequence to the 2nd tape such that our tapes become

1st tape	DNA_sequence#target_sequence
2nd tape	DNA_sequence

2° Non-deterministically check whether DNA_sequence includes target_sequence

to complete your 2-tape NTM with the following draft.



Note that you **cannot add any extra nodes and paths.** Moreover,

$$\begin{cases}\Gamma \to S \\ \Gamma \to L\end{cases}$$

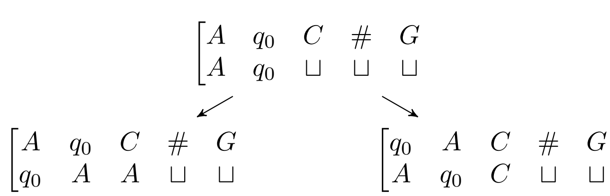
is represented as the transitions with all the combinations from $\Gamma \times \Gamma$:

$$\begin{cases}A \to S \\ A \to L\end{cases}, \ldots, \begin{cases}\sqcup \to S \\ A \to L\end{cases}, \begin{cases}A \to S \\ C \to L\end{cases}, \ldots, \begin{cases}\sqcup \to S \\ C \to L\end{cases}, \ldots, \begin{cases}A \to S \\ \sqcup \to L\end{cases}, \ldots, \begin{cases}\sqcup \to S \\ \sqcup \to L\end{cases}.$$

(b) (10 pts) Please simulate your 2-tape NTM on an input string

$$AG\#A$$

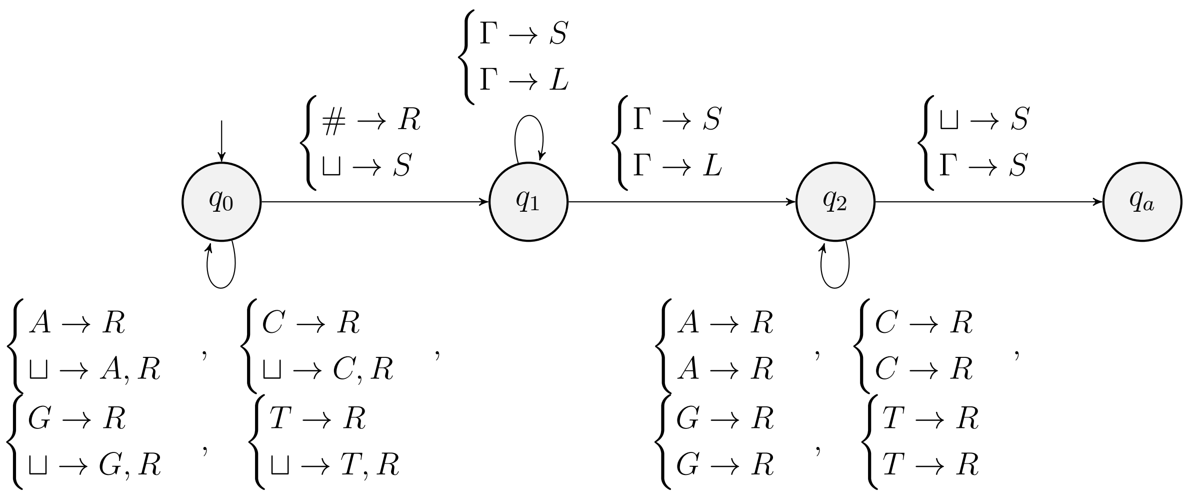
by drawing the corresponding simulation trees. Then, determine whether your NTM accepts the input string according to your simulation. A simulation tree is like the following example (as an illustration and not related to the NTM in this subproblem.)



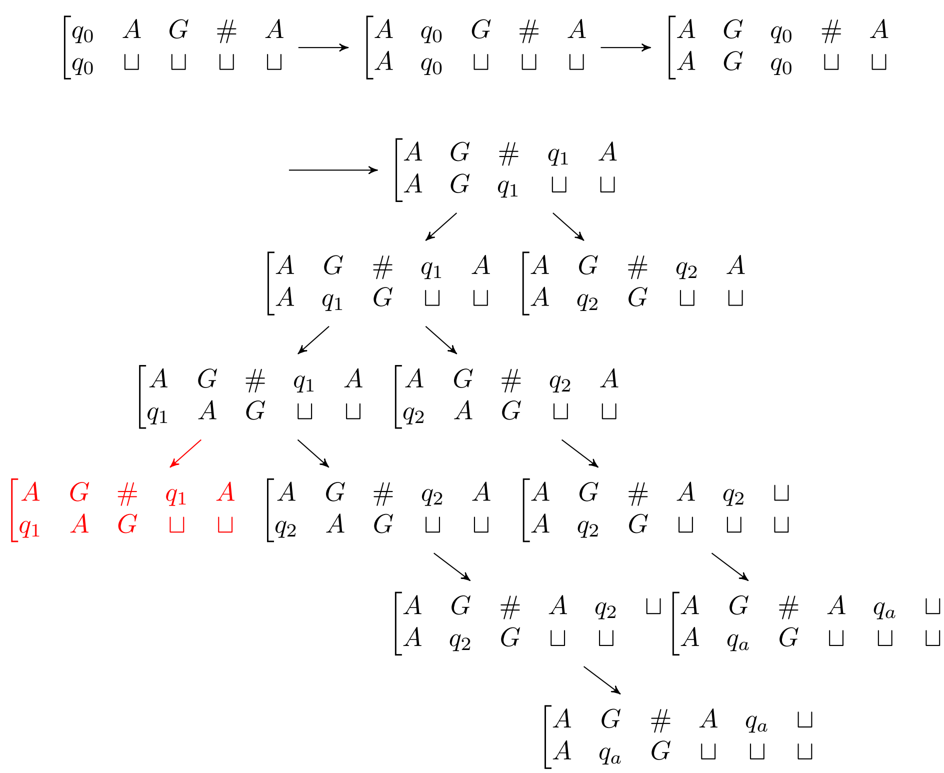
If the tree recursively contains the same branches during the simulation, please state the reason and do not continue the simulation on that branch.

Solution.

(a) Please see the following diagram



(b) Please see the following simulation



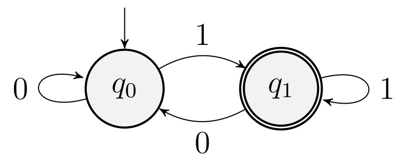
The red branch leads to an infinite loop, so we do not include it in the simulation. Because, by the definition of NTM, an input $\boldsymbol{w}$ is considered accepted if one branch work (in page 2 of the slide `chap3_NTM1.pdf`), we have reached the accept state q_a. Thus, this 2-tape NTM accepts the input string

$$AG\#A.$$

Problem 5 (25 pts). In our slides of Chapter 4, you learned that a TM can simulate a DFA on a string $\boldsymbol{w}$. Now, let us consider the DFA B:

$(Q, \Sigma, \delta, q_0, F) = (\{q_0, q_1\}, \{0, 1\}, \delta, q_0, \{q_1\})$,
where the transition function is

$$\begin{aligned} \delta: \ Q \times \Sigma &\rightarrow Q \\ (q, c) &\mapsto \tilde{q} \end{aligned}$$



To encode δ function into a string, we re-format $\delta(q, c) = \tilde{q}$ as $qc\tilde{q}$. That is, $\{q_0 0 q_0, q_0 1 q_1, q_1 0 q_0, q_1 1 q_1\}$. Therefore, we can simulate the DFA B on $\boldsymbol{w}$ via a TM with the input

$$(\{q_0, q_1\}, \{0, 1\}, \{q_0 0 q_0, q_0 1 q_1, q_1 0 q_0, q_1 1 q_1\}, q_0, \{q_1\})\#\boldsymbol{w}.$$

To simplify the problem, we remove the notations "(", ")", "{", "}" and ",", and re-define the input as

$$q_0 q_1 \mid 01 \mid q_0 0 q_0 q_0 1 q_1 q_1 0 q_0 q_1 1 q_1 \mid q_0 \mid q_1 \# \boldsymbol{w}, \tag{1}$$

where the notation "|" partitions the formal definition, and the elements of a set are concatenated.

(a) (5 pts) First, let us check whether $\boldsymbol{w} \in \Sigma^*$. Specifically, if $\boldsymbol{w}$ is "001", the TM accepts the input

$$q_0 q_1 \mid 01 \mid q_0 0 q_0 q_0 1 q_1 q_1 0 q_0 q_1 1 q_1 \mid q_0 \mid q_1 \# 001 \sqcup.$$

As a rejected example, if $\boldsymbol{w}$ is "528", the TM rejects the input

$$q_0 q_1 \mid 01 \mid q_0 0 q_0 q_0 1 q_1 q_1 0 q_0 q_1 1 q_1 \mid q_0 \mid q_1 \# 528 \sqcup.$$

Please design a **decidable single-tape** TM to perform this check with the input (1) via

- $Q_{\text{TM}} = \{q_0^{\text{C}}, q_1^{\text{C}}, q_a^{\text{C}}, q_r^{\text{C}}\}$, $\Sigma_{\text{TM}} = \{q_0, q_1, 0, 1, |, \#, 2, \cdots 9\}$ and $\Gamma = \{q_0, q_1, 0, 1, |, \#, \sqcup, 2, \cdots 9\}$.
- Let q_0^{C} be the start state, q_a^{C} be the accepted state and q_r^{C} be the rejected state.
- We only consider moving the head **right** or **left** in the TM.

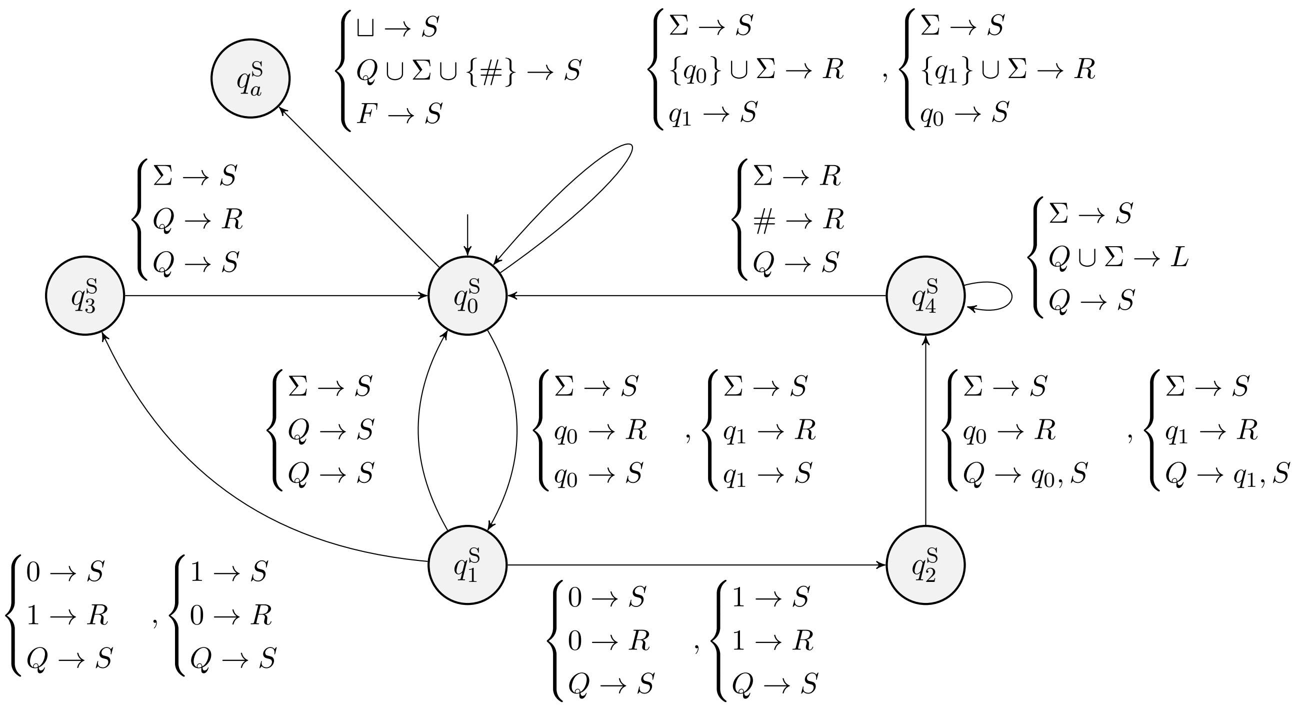
(b) (10 pts) Now, suppose that we have checked that B is a valid DFA. In the next step, let us simulate $\boldsymbol{w}$ according to the encoded δ part and check whether B accepts it. However, a decidable TM is too complex in this sub-problem, so we only focus on a **Turing-recognizable** TM here. Let us use a 3-tape TM to achieve the simulation, and the initial status of the tapes is

1st tape	$\# \ \mathbf{q_0^S} w_0 w_1 w_2 w_3 \cdots$
2nd tape	$\# \ \mathbf{q_0^S} q_0 0 q_0 q_0 1 q_1 q_1 0 q_0 q_1 1 q_1 \sqcup \cdots$
3rd tape	$\# \ \mathbf{q_0^S} q_0 \sqcup \cdots$

where we have # in the first position of each tape to denote the start position in the tape. Please simulate the input string $\boldsymbol{w} = 1$, i.e., the 1st tape is

$$\text{1st tape} \quad \# \ \mathbf{q_0^S} 1 \sqcup \cdots$$

with the following diagram. Note that you can skip the actions of self-loops on q_0^{S} and q_4^S, but you still need to draw the first and the last one.



(c) (10 pts) Continuing from problem (b), the idea behind the diagram is to sequentially determine the next state via 2nd tape's δ function, 3rd tape's current state, and 1st tape's current input w_i. Please describe the meaning of each state by completing the parts (I) $\cdots$ (VII) below:

State q_0^S: ___(I)___.

State q_1^S: Handle three possible situations when finding the current state in the δ function.

To State q_0 : ___(II)___, to State q_2 : ___(III)___, to State q_3 : ___(IV)___.

State q_2^S: ___(V)___.

State q_3^S: ___(VI)___.

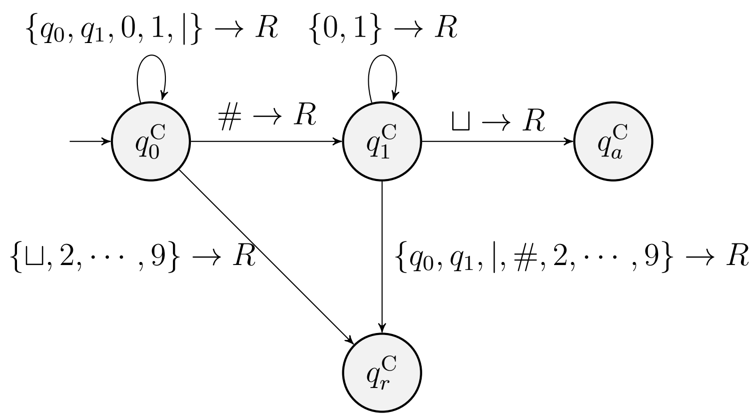
State q_4^S: Move left to the leftmost position in the δ function, and then move to the next input w_{i+1}.

State q_a^S: ___(VII)___.

Please note that if you want to reference the current input explicitly, you can use w_i.

Solution.

(a) Please see the following diagram.



(b) Here is the simulation. We ignore

$$
\begin{bmatrix} \# & \mathbf{q_0^S} & 1 & \sqcup & \cdots \\ \# & \mathbf{q_0^S} & q_0 & 0 & q_0q_01q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_0^S} & q_0 & \sqcup & \cdots \end{bmatrix}
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_1^S} & 1 & \sqcup & \cdots \\ \# & q_0 & \mathbf{q_1^S} & 0 & q_0q_01q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_1^S} & q_0 & \sqcup & \cdots \end{bmatrix}
$$

$$
\Rightarrow
\begin{bmatrix} \# & q_3^S & 1 & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_3^S & q_0 & q_01q_1q_10q_0q_11q_1 \\ \# & q_3^S & q_0 & \sqcup & \sqcup & \cdots \end{bmatrix}
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_0^S} & 1 & \sqcup & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_0 & \mathbf{q_0^S} & q_0 & 1q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_0^S} & q_0 & \sqcup & \sqcup & \sqcup & \cdots \end{bmatrix}
$$

$$
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_1^S} & 1 & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_0 & q_0 & \mathbf{q_1^S} & 1 & q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_1^S} & q_0 & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \end{bmatrix}
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_2^S} & 1 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_0 & q_0 & 1 & \mathbf{q_2^S} & q_1 & q_10q_0q_11q_1 \\ \# & \mathbf{q_2^S} & q_0 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \end{bmatrix}
$$

$$
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_4^S} & 1 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_0 & q_0 & 1 & q_1 & \mathbf{q_4^S} & q_1 & 0q_0q_11q_1 \\ \# & \mathbf{q_4^S} & q_1 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \end{bmatrix}
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_4^S} & 1 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \\ \# & q_0 & 0 & q_0 & q_0 & 1 & \mathbf{q_4^S} & q_1 & q_10q_0q_11q_1 \\ \# & \mathbf{q_4^S} & q_1 & \sqcup & \sqcup & \sqcup & \sqcup & \sqcup & \cdots \end{bmatrix}
$$

$\Rightarrow$ Move left (self loop on q_4) until reaching #

$$
\Rightarrow
\begin{bmatrix} \# & \mathbf{q_4^S} & 1 & \sqcup & \cdots \\ \mathbf{q_4^S} & \# & q_0 & 0 & q_0q_01q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_4^S} & q_1 & \sqcup & \cdots \end{bmatrix}
$$

$$
\Rightarrow
\begin{bmatrix} \# & 1 & \mathbf{q_0^S} & \sqcup & \cdots \\ \# & \mathbf{q_0^S} & q_0 & 0 & q_0q_01q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_0^S} & q_1 & \sqcup & \cdots \end{bmatrix}
\Rightarrow
\begin{bmatrix} \# & 1 & \mathbf{q_a^S} & \sqcup & \cdots \\ \# & \mathbf{q_a^S} & q_0 & 0 & q_0q_01q_1q_10q_0q_11q_1 \\ \# & \mathbf{q_a^S} & q_1 & \sqcup & \cdots \end{bmatrix}
$$

(c) The parts (I) $\cdots$ (VII):

(I) If the entire $\boldsymbol{w}$ has not been read yet, find the state of the encoded transition function that matches the current state on the third tape. Otherwise, accept it if the DFA B accepts it.

(II) Because we mistakenly treated the output of the transition function as the input(i.e.,$\tilde{q}$ of $qc\tilde{q}$), we need to address this. Therefore, skip it and look for the next $qc\tilde{q}$.

(III) Current input w_i **match** the input alphabet of transition function(i.e., c of $qc\tilde{q}$).

(IV) Current input w_i **did not match** the input alphabet of transition function(i.e., c of $qc\tilde{q}$).

(V) Update the current state in 3rd tape when encountering the situation (III).

(VI) Move two steps to the right to the next $qc\tilde{q}$ when encountering the situation (IV).

(VII) Accept if the entire input $\boldsymbol{w}$ has been read and the current state is in accept states of DFA B.