Middle East Technical University Department of Computer Engineering

CENG 280

Formal Languages and Abstract Machines

Spring 2018-2019

Take Home Exam 3

Due date: May 22, 2019

Objectives

To familiarize with computation using Turing Machines and inspect aspects of the class of languages associated with them.

Specifications

You must adhere to the notation conventions adapted in the textbook. Use the standard deterministic Turing Machine as defined in the book unless stated otherwise. You should utilize basic machines notation whenever asked for.

Your solution should be delivered as a `pdf` file generated using the `tex` template provided or employing other text editors. You are allowed to use only text and vector images in your submission, i.e. use of raster images is clearly forbidden.

Questions and submission regulations are included in subsequent sections. Assume that any input string w over a finite alphabet Σ is initially placed on the tape of the standard TM always as $\triangleright\sqcup w$ unless stated otherwise. If multi-tape TM is used then the same placement occurs on its first tape and initial configuration will be as $\triangleright\sqcup$ concerning the remaining tapes.

Designing your solutions to the tasks, explicitly state any assumptions you make and pay particular attention to the notation you use. Your proofs must be sound and complete. Grading will be heavily affected by the formalization of your solutions.

Question 1 (12 pts)

Answer the questions using the following Turing Machine.

$$> R \xrightarrow{a \neq \sqcup} \sqcup L_{\sqcup}\, a$$

a. Write the given TM in **five-tuple notation** as $M = (K, \Sigma, \delta, s, H)$ **explicitly** specifying each constituent. Let $\Sigma = \{a, b, \sqcup, \triangleright\}$ and $\delta(q, \triangleright) = (q, \rightarrow)$ for all $q \in K - H$. Denote δ as a **table**. (6 pts)

$M = (K,\ \Sigma,\ \delta,\ s,\ H)$ where $K = \{s,\ q_1,\ q_{a1},\ q_{a2},\ q_{a3},\ q_{b1},\ q_{b2},\ q_{b3},\ h_1,\ h_2,\ h_3\}$, $\Sigma = \{a,\ b\}$, $H = \{h_1,\ h_2,\ h_3\}$, and δ is depicted in the table below.

$\boldsymbol{q}$	$\boldsymbol{\sigma}$	$\boldsymbol{\delta(q, \sigma)}$
s	a	$(q_1, \rightarrow)$
s	b	$(q_1, \rightarrow)$
s	$\sqcup$	$(q_1, \rightarrow)$
s	$\triangleright$	$(s, \rightarrow)$
q_1	a	$(q_{a1}, \sqcup)$
q_1	b	$(q_{b1}, \sqcup)$
q_1	$\sqcup$	$(h_1, \sqcup)$
q_1	$\triangleright$	$(q_1, \rightarrow)$
q_{a1}	a	$(q_{a2}, \leftarrow)$
q_{a1}	b	$(q_{a2}, \leftarrow)$
q_{a1}	$\sqcup$	$(q_{a2}, \leftarrow)$
q_{a1}	$\triangleright$	$(q_{a1}, \rightarrow)$
q_{a2}	a	$(q_{a2}, \leftarrow)$
q_{a2}	b	$(q_{a2}, \leftarrow)$
q_{a2}	$\sqcup$	$(q_{a3}, \sqcup)$
q_{a2}	$\triangleright$	$(q_{a2}, \rightarrow)$
q_{a3}	a	(h_2, a)
q_{a3}	b	(h_2, a)
q_{a3}	$\sqcup$	(h_2, a)
q_{a3}	$\triangleright$	$(q_{a3}, \rightarrow)$
q_{b1}	a	$(q_{b2}, \leftarrow)$
q_{b1}	b	$(q_{b2}, \leftarrow)$
q_{b1}	$\sqcup$	$(q_{b2}, \leftarrow)$
q_{b1}	$\triangleright$	$(q_{b1}, \rightarrow)$
q_{b2}	a	$(q_{b2}, \leftarrow)$
q_{b2}	b	$(q_{b2}, \leftarrow)$
q_{b2}	$\sqcup$	$(q_{b3}, \sqcup)$
q_{b2}	$\triangleright$	$(q_{b2}, \rightarrow)$
q_{b3}	a	(h_3, b)
q_{b3}	b	(h_3, b)
q_{b3}	$\sqcup$	(h_3, b)
q_{b3}	$\triangleright$	$(q_{b3}, \rightarrow)$

b. Using M defined in **(a)**, write down the **computation** (sequence of configurations related by $\vdash_M$) for the following initial configurations. (6 pts)

(i) $(s, \rhd \sqcup \sqcup b\underline{a}b)$.

$$\begin{aligned}
(s, \rhd \sqcup \sqcup b\underline{a}b) &\vdash_M (q_1, \rhd \sqcup \sqcup ba\underline{b})\\
&\vdash_M (q_{b1}, \rhd \sqcup \sqcup ba\underline{\sqcup})\\
&\vdash_M (q_{b2}, \rhd \sqcup \sqcup b\underline{a})\\
&\vdash_M (q_{b2}, \rhd \sqcup \sqcup \underline{b}a)\\
&\vdash_M (q_{b2}, \rhd \sqcup \underline{\sqcup}ba)\\
&\vdash_M (q_{b3}, \rhd \sqcup \underline{\sqcup}ba)\\
&\vdash_M (h_3, \rhd \sqcup \underline{b}ba).
\end{aligned}$$

(ii) $(s, \rhd a\underline{a}a)$.

$$\begin{aligned}
(s, \rhd a\underline{a}a) &\vdash_M (q_1, \rhd aa\underline{a})\\
&\vdash_M (q_{a1}, \rhd aa\underline{\sqcup})\\
&\vdash_M (q_{a2}, \rhd a\underline{a})\\
&\vdash_M (q_{a2}, \rhd \underline{a}a)\\
&\vdash_M (q_{a2}, \underline{\rhd}aa)\\
&\vdash_M (q_{a2}, \rhd \underline{a}a)\\
&\vdash_M (q_{a2}, \underline{\rhd}aa)\\
&(...)\\
&\text{does not halt.}
\end{aligned}$$

(iii) $(s, \rhd \underline{a} \sqcup bb)$.

$$\begin{aligned}
(s, \rhd \underline{a} \sqcup bb) &\vdash_M (q_1, \rhd a\underline{\sqcup}bb)\\
&\vdash_M (h_1, \rhd a\underline{\sqcup}bb).
\end{aligned}$$

Question 2 (8 pts)

Given the following Turing Machine on the alphabet $\Sigma = \{a, b, c, \sqcup, \rhd\}$,

$$> R \xrightarrow{a \neq \sqcup} R_{\sqcup}\, L \xrightarrow{b = c} R \sqcup R\, a\, b\, R \sqcup R\, a$$

trace its operation on the input string $babc$ (intially placed on the tape as $\rhd\underline{\sqcup}babc$) without explicitly referring to the particular set of states. In plain English, write down the set of **actions taken** by the TM upon encountering **any** string over $(\Sigma - \{\sqcup, \rhd\})^*$ on its tape.

$$\begin{aligned}
&\rhd\underline{\sqcup}babc \vdash \rhd \sqcup \underline{b}abc \vdash \rhd \sqcup babc\underline{\sqcup} \vdash \rhd \sqcup bab\underline{c} \vdash \rhd \sqcup babc\underline{\sqcup} \vdash \rhd \sqcup babc\underline{\sqcup} \vdash \rhd \sqcup babc \sqcup \underline{\sqcup} \vdash \rhd \sqcup babc \sqcup \underline{b}\\
&\vdash \rhd \sqcup babc \sqcup \underline{c} \vdash \rhd \sqcup babc \sqcup c\underline{\sqcup} \vdash \rhd \sqcup babc \sqcup c \sqcup \underline{\sqcup} \vdash \rhd \sqcup babc \sqcup c \sqcup \underline{b}.
\end{aligned}$$

If the non-empty input string over $\{a, b, c\}^*$ has length at least two and ends with c, the machine goes to the end of the string writing a blank followed by the last and the first characters in the string separated by another blank. If the input string is equal to c the machine writes two c's in the mentioned format to the portion of the tape next to the end of the input.

Question 3 (10 pts)

Given the following Turing Machine M on the alphabet $\Sigma = \{a, b, \#, \sqcup, \triangleright\}$, answer the questions.

$$\begin{tikzcd}
> R_{\overline{a}} \arrow[r, "b"] \arrow[dr, bend right, "\sqcup"'] & R_{\overline{b}} \arrow[r, "a"] \arrow[d, "\sqcup"] & \# \arrow[r] & L_{\overline{b}} R\, a\, R_{\#} b \arrow[r] & R \arrow[lll, bend right=40, "b"'] \arrow[ll, bend left=40, "a"] \arrow[dlll, bend left=30, "\sqcup"] \\
& L_{\sqcup} & & &
\end{tikzcd}$$

a. Write the language L over $\{a, b\}$ that M **semidecides** using set notation. (5 pts)

$L = \{w : w \in \{a, b\}^*\}$.

b. Using **Definition 4.2.2** of the textbook, identify the function $f : \{a, b\}^* \mapsto \{a, b\}^*$ such that $M(w) = f(w)$ where $w \in \{a, b\}^*$. (5 pts)

$M(w) = f(w) = a^{n_a(w)} b^{n_b(w)}$, where n_a and n_b are functions from $\{a, b\}^*$ to $\mathbb{N}$ mapping inputs to the number of a's and b's they contain, respectively.

Question 4 (10 pts)

Using **basic machines notation**, construct a **multi-tape** Turing Machine that **decides** the following language:

$L = \{wcu : w, u \in \{a, b\}^+, \; w(2i) = w(2i-1) = u(i) \text{ for } i = 1, ..., |u|, \; |w| = 2|u|\}$.

$$\begin{tikzcd}
> R^{1,2} \arrow[r, "z^1 = a, b"] \arrow[ddr, bend right, "\sqcup^1,\, c^1"'] & x^2 \arrow[r] & R^1 \arrow[r, "p^1 = z^1"] \arrow[ddl, bend right, "p^1 \neq z^1"'] & R^{1,2} \arrow[r, "c^1"] \arrow[ddll, "\sqcup^1"'] \arrow[d, "x^1 = a, b"] & L^2_{\sqcup} \arrow[dd] & \\
& & & x^2 \arrow[d] & & \\
& \mathbf{n} & & R^1 \arrow[uu, bend left, "r^1 = x^1"] \arrow[ll, bend left=10, "r^1 \neq x^1"'] & R^{1,2} \arrow[loop right, "s^1 \neq \sqcup,\, s^2 \neq \sqcup,\, s^1 = s^2"] \arrow[lll, bend left=30, "s^1 \neq s^2"] \arrow[dr, bend right, "\sqcup^1 \sqcup^2"'] & \\
& & & & & \mathbf{y}
\end{tikzcd}$$

Question 5 (10 pts)

Using **basic machines notation**, construct a **multi-tape** Turing Machine that **computes** the following function:

$f\ :\ \{a,b\}^* \mapsto \{a,b\}^*$ where $f(e) = e$ and $f(u) = v$ where $u, v \in \{a,b\}^+$ and $|v| \leq |u| + 1$ such that

$$v(2i-1) = u(i)$$
$$v(2i) = u(|u| - i + 1)$$

for $i \in \mathbb{N}$ within $\left[1, \left\lceil \frac{|u|}{2} \right\rceil \right]$.

$$\begin{tikzcd}
> R^{1,2} \arrow[r, "x^1\ \neq \sqcup"] \arrow[d, "\sqcup^1"] & \#^1 \arrow[rr] & & x^2 R^1_\sqcup \arrow[r] & L^1_\$ \arrow[dll, bend right, "z^1 \neq \#"'] \arrow[dr, bend left, "z^1 = \#"] & \\
L^1_\sqcup & & \$^1 \arrow[r] & R^2 \arrow[r] & z^2 \arrow[d] & R^2 \arrow[d] \\
& & & R^1 \arrow[uull, bend left, "\overline{\$}^1"] \arrow[d, "\$^1"] & L^1_\# \arrow[l] & x^2 \arrow[dll, bend left] \\
s^1 \arrow[r, bend right] & R^{1,2} \arrow[l, bend right, "s^2\ \neq \sqcup"'] \arrow[uul, bend right, "\sqcup^2"'] & & L^{1,2}_\sqcup \arrow[ll] & &
\end{tikzcd}$$

Question 6 (12 pts + 10 pts bonus)

A variant of Turing Machine is the deterministic **queue TM** defined as a quintuple $(K, \Sigma, \delta, s, H)$ where K is the set of states; Σ is a finite alphabet containing $\triangleright$ to denote the left end of the queue; $s \in K$ is the initial state; $H \subseteq K$ is the set of halting states; and $\delta\ :\ (K - H) \times \Sigma \mapsto K \times (\Sigma \cup \{\downarrow\})$ is the transition function where $\delta(q, a) = (p, X)$ for $q \in K - H$, $p \in K$, $a \in \Sigma$, $X \in \Sigma \cup \{\downarrow\}$ indicates that whenever the machine is in state q and there is a at the **front** position of the queue, the machine enters the state p and takes the action X. If $X \in \Sigma$, the machine **enqueues** the element X at the **rear** position of the queue. Otherwise if $X = \downarrow$, the machine **pops** the element at the front. Note that by these mechanisms the queue can get arbitrarily large or small.

a. Impose restrictions on δ so that the pop operation is **not** allowed in an empty queue and enqueueing of $\triangleright$ **is** forbidden. (4 pts)

For all $q \in K - H$, $p \in K$, $a \in \Sigma$, $b \in \Sigma$, $X \in \Sigma \cup \{\downarrow\}$;

If $\delta(q, \triangleright) = (p, X)$ then $X \neq \downarrow$.

If $\delta(q, a) = (p, b)$ then $b \neq \triangleright$.

b. Add the flexibility of taking action **regardless of** what element is placed at the front of the queue, using e-transitions in δ in a way that **preserves** the determinism of the machine. Also update the sets over which δ is defined. (4 pts)

Partition $K = K_e \cup K_{\overline{e}} \cup H$ such that $K_e \cap K_{\overline{e}} = \emptyset$, $K_{\overline{e}} \cap H = \emptyset$, and $K_e \cap H = \emptyset$.
Then, δ is now a mapping from $(K - H) \times (\Sigma \cup \{e\})$ to $K \times (\Sigma \cup \{\downarrow\})$ with the following restrictions.

For all $q \in K_e$, $p \in K$, $a \in \Sigma - \{\triangleright\}$, $b \in (\Sigma - \{\triangleright\}) \cup \{\downarrow\}$; if $\delta(q, a) = (p, b)$, then $a = e$; and regardless of what element is at the front action b is taken. Note that we still have to define $\delta(q, \triangleright)$ transitions in accordance with the restrictions put forward in the previous option.

For all $q \in K_{\overline{e}}$ and $a \in \Sigma$, $\delta(q, a)$ must be uniquely defined to ensure that δ is a function (of course).

c. Define the **yields-in-one-step** relation of the machine, covering every case. (4 pts)

Assume that a configuration of a non-empty queue TM is represented by a tuple $(q, \triangleright xw)$ s.t. $q \in K$, $x \in \Sigma - \{\triangleright\}$ and $w \in \Sigma - \{\triangleright\}^*$ where x is the element at the front position. Moreover, let $(q, \triangleright)$ depict an empty queue TM configuration.

Yields-in-one-step relation $\vdash_M$ is a subset of the cross product of the queue TM M's configurations under the following rules. Let $q \in K - H$, $p \in K$, $a \in \Sigma - \{\triangleright\}$, $b \in \Sigma - \{\triangleright\}$, and $w \in (\Sigma - \{\triangleright\})^*$.

$(q, \triangleright) \vdash_M (p, \triangleright a)$ if $\delta(q, \triangleright) = (p, a)$.

$(q, \triangleright a) \vdash_M (p, \triangleright)$ if $\delta(q, a) = (p, \downarrow)$.

$(q, \triangleright aw) \vdash_M (p, \triangleright awb)$ if $\delta(q, a) = (p, b)$.

$(q, \triangleright awb) \vdash_M (p, \triangleright wb)$ if $\delta(q, a) = (p, \downarrow)$.

d. Construct a deterministic queue TM to **semidecide** the language $L = \{w^R cw : w \in \{a, b\}^*\}$. Your solution to this option will be considered as a bonus, provide a solution if you want to get extra points. You may invent your own graphical notation of the queue TM. (bonus: 10 pts)

Question 7 (20 pts)

Another variant of the Turing Machine can be defined as a deterministic TM where you can read the character at the **front** and **rear** positions of the input string on the tape, and perform **insertion** of new cells initialized with a character in the alphabet and **deletion** of existing cells **only** at the front and rear positions. Call this TM the **insert-delete TM** and assume that it does not have any additional functionality other than those mentioned.

a. Define the insert-delete TM as a **tuple**, stating the type of each constituent and identifying their functionalities. (5 pts)

Insert-delete TM is a quintuple $(K, \Sigma, \delta, s, H)$ where K is a finite set of states, Σ is a finite alphabet containing the left end symbol $\triangleright$, $s \in K$ is the start state, $H \subseteq K$ is the set of halting states, and $\delta : (K - H) \times \Sigma \times \Sigma \mapsto K \times (\Sigma_F \cup \Sigma_R \cup \{\downarrow_f, \downarrow_r\})$ where $\Sigma_F = \{a_F : a \in \Sigma - \{\triangleright\}\}$ and $\Sigma_R = \{a_R : a \in \Sigma - \{\triangleright\}\}$ is the state transition function such that for $a, b \in \Sigma$, $q \in K - H$, $p \in H$, $\delta(q, a, b) = (p, X)$ implies that whenever the machine is in state q, reading a at the front and b at the rear, it transitions into state p and carries out the action X. If $X \in \Sigma_F$, then a new cell with corresponding symbol in $\Sigma - \{\triangleright\}$ is inserted at the front position. Similarly, if $X \in \Sigma_R$, then a new cell with corresponding symbol in $\Sigma - \{\triangleright\}$ is inserted at the rear. These corresponding symbols can be found using one-to-one and onto functions $g_F : \Sigma - \{\triangleright\} \mapsto \Sigma_F$ and $g_R : \Sigma - \{\triangleright\} \mapsto \Sigma_R$ where $g_F(\sigma) = \sigma_F$ and $g_R(\sigma) = \sigma_R$ for $\sigma \in \Sigma - \{\triangleright\}$, $\sigma_F \in \Sigma_F$, $\sigma_r \in \Sigma_R$. Otherwise, if $X =\downarrow_F$, the cell at the front is deleted and if $X =\downarrow_R$, the cell at the rear is deleted. Similar to the queue TM in the previous question, inserting cells with $\triangleright$ values and deleting cells in an insert-delete TM with empty tape is forbidden.

b. Define the notion of **configuration** for the insert-delete TM. (2 pts)

Any member of $K \times \Sigma_\theta^*$ where $\Sigma_\theta = \Sigma \cup \{\underline{a} : a \in \Sigma\} \cup \{\overline{a} : a \in \Sigma\} \cup \{\underline{\overline{a}} : a \in \Sigma\}$. For $a \in \Sigma$, $\overline{a}$ indicates that a is at the front end, $\underline{a}$ indicates that a is at the back end, and $\underline{\overline{a}}$ indicates that both front and rear ends point to a at the same time. Empty string placed on tape at $q \in K$ should be represented as $(q, \underline{\overline{\triangleright}})$.

c. Define the **yields-in-one-step** relation of the machine. Write in set notation the language **recognized** by the reflexive transitive closure of yields-in-one-step relation. (4 pts)

Following types of configurations are related to each other via $\vdash_M$ pertaining to an insert-delete TM $M = (K, \Sigma, \delta, s, H)$. Assume that $q \in K - H$, $p \in K$, $a \in \Sigma - \{\triangleright\}$, $b \in \Sigma - \{\triangleright\}$, $c \in \Sigma - \{\triangleright\}$, $w \in (\Sigma - \{\triangleright\})^*$, $a_F \in \Sigma_F$, $a_R \in \Sigma_R$, $b_F \in \Sigma_F$, $b_R \in \Sigma_R$, $c_F \in \Sigma_F$, and $c_R \in \Sigma_R$.

$(q, \underline{\overline{\triangleright}}) \vdash_M (p, \triangleright\underline{\overline{a}})$ if $\delta(q, \triangleright, \triangleright) = (p, a_F)$ or $\delta(q, \triangleright, \triangleright) = (p, a_R)$.

$(q, \triangleright\underline{\overline{a}}) \vdash_M (p, \underline{\overline{\triangleright}})$ if $\delta(q, a, a) = (p, \downarrow_F)$ or $\delta(q, a, a) = (p, \downarrow_R)$.

$(q, \triangleright\underline{\overline{a}}) \vdash_M (p, \triangleright\overline{a}\underline{b})$ if $\delta(q, a, a) = (p, b_R)$.

$(q, \triangleright\underline{\overline{a}}) \vdash_M (p, \triangleright\overline{b}\underline{a})$ if $\delta(q, a, a) = (p, b_F)$.

$(q, \triangleright\overline{a}\underline{b}) \vdash_M (p, \triangleright\underline{\overline{b}})$ if $\delta(q, a, b) = (p, \downarrow_F)$.

$(q, \triangleright\overline{a}\underline{b}) \vdash_M (p, \triangleright\underline{\overline{a}})$ if $\delta(q, a, b) = (p, \downarrow_R)$.

$(q, \triangleright\overline{a}w\underline{b}) \vdash_M (p, \triangleright\overline{a}wb\underline{c})$ if $\delta(q, a, b) = (p, c_R)$.

$(q, \triangleright\overline{a}w\underline{b}) \vdash_M (p, \triangleright\overline{c}aw\underline{b})$ if $\delta(q, a, b) = (p, c_L)$.

$(q, \triangleright\overline{a}cw\underline{b}) \vdash_M (p, \triangleright\overline{c}w\underline{b})$ if $\delta(q, a, b) = (p, \downarrow_L)$.

$(q, \triangleright\overline{a}wc\underline{b}) \vdash_M (p, \triangleright\overline{a}w\underline{c})$ if $\delta(q, a, b) = (p, \downarrow_R)$.

$L(M) = \{w \in (\Sigma - \triangleright)^* : (s, \Phi(w)) \vdash_M^* (h, u),\ h \in H, u \in \Sigma^*\}$ is the language recognized by M, where $\Phi : (\Sigma - \triangleright)^* \mapsto \Sigma_\theta^*$ such that $\Phi(e) = \underline{\overline{\triangleright}}$, $\Phi(a) = \triangleright\underline{\overline{a}}$, $\Phi(avb) = \triangleright\overline{a}v\underline{b}$ for $a, b \in \Sigma - \triangleright$ and $v \in (\Sigma - \triangleright)^*$.

d. Prove that the insert-delete TM is **equivalent** to the standard TM. Write down how each case should be proved without going over every possible detail. (9 pts)

We will only consider the most general configurations where strings on the tapes are sufficiently large. Focus will be on how we can reproduce the functionality of one machine using the other.

PART I: Any computation of the standard TM can be emulated by the insert-delete TM.
Assume that a configuration of a standard TM is given as $\triangleright\alpha_1...\alpha_m\underline{a}\beta_1...\beta_n$ with the state abstracted away where m and n are sufficiently large natural numbers, and all the used variables are members of the machine's alphabet. So as to represent this configuration in the insert-delete TM, we have to place a either at the front or at the rear since only through these positions we can manipulate tape contents by definition. I choose to use the front opening and given configuration is represented as $\triangleright\overline{a}\beta_1...\beta_n\#\alpha_1...\underline{\alpha_m}$ where right portion and left portion of the input on the tape are separated by $\#$.

- To write a symbol b to the square the head points to on the standard TM, delete the cell at the front of the insert-delete TM and insert a new cell with a value b once more at the front position. End configuration of the standard TM becomes $\triangleright\alpha_1...\alpha_m\underline{b}\beta_1...\beta_n$ while the insert-delete TM configuration transforms into $\triangleright\overline{b}\beta_1...\beta_n\#\alpha_1...\underline{\alpha_m}$.

- To go to one square left of where the head points to on the standard TM, read the value at the rear, then delete that cell and insert a new cell with the read value at the front position in the insert-delete TM. End configuration of the standard TM becomes $\triangleright\alpha_1...\underline{\alpha_m}b\beta_1...\beta_n$ while the insert-delete TM configuration transforms into $\triangleright\overline{\alpha_m}a\beta_1...\beta_n\#\alpha_1...\underline{\alpha_{m-1}}$.

- To go to one square right of where the head points to on the standard TM, read the value at the front, then delete that cell and insert a new cell with the read value at the rear position in the insert-delete TM. End configuration of the standard TM becomes $\triangleright\alpha_1...\alpha_m b\underline{\beta_1}...\beta_n$ while the insert-delete TM configuration transforms into $\triangleright\overline{\beta_1}...\beta_n\#\alpha_1...\alpha_m\underline{a}$.

PART II: Any computation of the insert-delete TM can be emulated by the standard TM.
Given a configuration of the insert-delete TM as $\triangleright\overline{a}\sigma_1...\ \sigma_m\underline{b}$ where m is a sufficiently large positive integer and each variable denotes a character in the alphabet, we can represent this configuration on a two-tape TM as $(\triangleright\underline{a}\sigma_1...\sigma_m b,\ \triangleright\underline{b}\sigma_m...\ \sigma_1 a)$ so that the second tape contains the reversed string on the first tape.

- To delete the front cell in the insert-delete TM, we go to the leftmost character on the first tape of the two-tape TM, change it with $\sqcup$, and shift the string on that tape one square left. Regarding the second tape, we replace the last character of the string with $\sqcup$. End configuration of the insert-delete TM becomes $\triangleright\overline{\sigma_1}...\ \sigma_m\underline{b}$ while the two-tape TM configuration transforms into $(\triangleright\underline{\sigma_1}...\sigma_m b,\ \triangleright\underline{b}\sigma_m...\ \sigma_1)$.

- To delete the rear cell in the insert-delete TM, we change the last character on the first tape of the two-tape TM with $\sqcup$. Regarding the second tape, we go to the leftmost character, change it with $\sqcup$, and shift the remaining string one square left. End configuration of the insert-delete TM becomes $\triangleright\overline{a}\sigma_1...\ \underline{\sigma_m}$ while the two-tape TM configuration transforms into $(\triangleright\underline{a}\sigma_1...\sigma_m,\ \triangleright\underline{\sigma_m}...\sigma_1 a)$.

- To insert a new cell with value c at the front position of the insert-delete TM, we shift the string on the first tape of the two tape TM one square right, find the leftmost $\sqcup$ and write c at this position. Regarding the second tape, go the first $\sqcup$ after the end of the input and write c there. End configuration of the insert-delete TM becomes $\triangleright\overline{c}a\sigma_1...\sigma_m b$ while the two-tape TM configuration transforms into $(\triangleright\underline{c}a\sigma_1...\sigma_m b,\ \triangleright\underline{b}\sigma_m...\ \sigma_1 ac)$.

- To insert a new cell with value c at the rear position of the insert-delete TM, we go to the first $\sqcup$ following the input string on the first tape of the two-tape TM and write c at this position. Regarding the second tape, we shift the string on the tape one square right, find the leftmost $\sqcup$ and write c there. End configuration of the insert-delete TM becomes $\triangleright\overline{a}\sigma_1 ... \sigma_m b\underline{c}$ while the two-tape TM configuration transforms into $(\triangleright\underline{a}\sigma_1...\sigma_m bc, \triangleright\underline{c}b\sigma_m ... \sigma_1 a)$.

Since the two-tape TM is equivalent to the standard TM in terms of the equivalence of the sets of the languages they recognize as established by Theorem 4.3.1, we infer that any computation of the insert-delete TM can be emulated by the standard TM.

Question 8 (10 pts)

Construct an **unrestricted grammar** that **generates** the language

$L = \{a^{f(n)} \ : \ f(n) = \Sigma_{i=0}^{n} i^2, \, n \in \mathbb{N}\}$.

(Note that $n^2 \ = \ 1 \ + \ 3 \ + \ ... \ + \ (2n-1)$ for $n \in \mathbb{N} - \{0,1\}$.)

$G = (V, \Sigma, R, S)$

where

$V = \{a, S, L, R, C, \$, \#, H, Q, T, B, N, Y, Z, F\}$,
$\Sigma = \{a\}$,

$R = \{S \to e,$	initial
$S \to LRaR,$	initial

$LRa \to LRCa,$	initiate copy
$Ca \to AaC,$	copy a as A
$C\# \to \$\#C,$	copy # as $
$aCR \to aR,$	end copy
$aA \to Aa,$	rearrange A to left of a
$\#\$ \to \$\#,$	rearrange $ to left of #
$a\$ \to \$a,$	rearrange a to right of $
$\#A \to A\#,$	rearrange A to left of #
$LRA \to LRHA,$	initiate copy check, order must be (A, $) blocks, then ($a$, #) blocks
$HA \to QH,$	replace A with Q
$H\$A \to \#QH,$	replace $ between A sequences with #
$QHa \to QRa,$	put R between borders of A and a to finish copying

$LRQ \to LRTQ,$	initiate copy leftmost block of Q
$TQ \to BaT,$	copy Q as Ba
$aB \to Ba,$	group B to left of a
$aTR \to aR,$	end copy at first a block of just one a
$aT\# \to a\#N,$	initiate post-process right portion
$NQ \to aN,$	replace all remaining Q with a, this part is DONE
$aNR \to aR,$	end post-cleaning

$LRB \to LRYB\#,$	initiate add two operator to B block
$\#B \to B\#,$	put # separator to right end of B block
$YB \to aY,$	replace B with a
$Y\# \to aa\#Z,$	at the right border add aa
$Za \to aZ,$	keep a blocks intact
$Z\# \to Z,$	delete rightmost # separator
$aZR \to aR,$	end of post-processing

$LRa \to LRFa,$	initiate finishing operator: delete all remaining markers but a
$LRF \to LF,$	
$LFa \to aF,$	
$aFa \to aaF,$	
$F\# \to F,$	
$aFRa \to aaF,$	
$aFR \to aF\}.$	

Question 9 (8 pts)

Let L_1, L_2, and L_3 be **recursively enumerable** languages. Prove that the language $L = (L_1L_2) \cap L_3$ is a recursively enumerable language by devising a **nondeterministic TM** that **semidecides** L utilizing TM's for L_1, L_2, and L_3.

As L_1, L_2 and L_3 are recursively enumerable languages, they must be semidecided by some TM's. Assume that M_1, M_2, M_3 are deterministic TM's such that $L(M_1) = L_1$, $L(M_2) = L_2$ and $L(M_3) = L_3$. Without loss of generality also let L_1, L_2, L_3 be defined over the same alphabet Σ.

Construct the nondeterministic TM M for L as follows:

On an input string $w \in \Sigma^*$:

1- Simulate w on M_3. If M_3 halts on w, go on with step 2.

2- Select w_1 and w_2 such that $w = w_1w_2$.

3- Simulate w_1 on M_1. If M_1 halts on w_1, proceed with step 4.

4- Simulate w_2 on M_2. If M_2 halts on w_2, halt.

Question 10 (not graded)

Prove that L is **not recursively enumerable** where

$L = \{$“M”“w” : “M” is encoding of a Turing machine and $w \notin L(M)\}$.

Submission

- You should submit your THE3 as a pdf file with the identifier `'the3.pdf'` on odtuclass. You may use the provided .tex template with appropriate modifications.
- Name your Turnitin submission in the format `<id>_<first-name>_<last-name>`.
- **Late Submission:** You have two days in total for late submission with penalties of 20 pts and 50 pts reduction in your grade for the first and second day, respectively. No further submissions are accepted.
- Do not submit solutions for not graded questions.

Regulations

1. **Cheating:** This take-home exam has to be completed and submitted **individually**. Teaming up, sharing solutions anywhere other parties might access and using work belonging to others as part or in whole are considered cheating. **We have zero tolerance policy for cheating**. People involved in cheating will be punished in accordance with the university regulations.
2. **Newsgroup:** You must follow the newsgroup (cow.ceng.metu.edu.tr) for discussions and possible updates on a daily basis. You are advised to initiate discussions on COW so that most parties will benefit.