FOUNDATIONS OF COMPUTATION 2019
PROBLEM CLASS 7

1. Turing machine for recognizing words of even length in $\Sigma_0 = \{a\}$. Let $Q = \{s, q\}$.

state	letter	δ(state,letter)
s	a	$(q,\ \rightarrow)$
q	a	$(s,\ \rightarrow)$
s	$\sqcup$	$(e,\ \sqcup)$
q	$\sqcup$	$(o,\ \sqcup)$

Where $e, o \in H$, and e indicates that the size of the input is even while o that it's odd. An example of a computation for this TM:

$$(s, \triangleright\underline{a}aa), (q, \triangleright a\underline{a}a), (s, \triangleright aa\underline{a}), (q, \triangleright aaa\underline{\sqcup}), (o, \triangleright aaa\underline{\sqcup}).$$

2. Turing machine with the input alphabet $\Sigma_0 = \{a, b\}$ such that given any word $w \in \Sigma^*$, it will change every a in w to a b, every b to an a, and then will halt.

The machine is $(Q, \Sigma, \delta, s, \{h\})$ where $Q = \{s, q, h\}$, h is a halting state, and δ is given by:

- $(s, \sqcup) \Rightarrow (h, \sqcup)$,
- $(s, a) \Rightarrow (q, b)$,
- $(s, b) \Rightarrow (q, a)$,
- $(q, \sqcup) \Rightarrow$ anything
- $(q, a) \Rightarrow (s, \rightarrow)$,
- $(q, b) \Rightarrow (s, \rightarrow)$.

3. Describe on implementation level a Turing machine which *decides* the language $\{w \in \{a, b, c\} |\ w$ contains an equal number of letters $a,\ b$ and $c\}$.

TM works in three stages.

Stage 1. Being in the *initial* state q_c, TM scans input from left to right. If the input does not contain any letters a, b, c, then then TM terminates with **Yes**. If the input does contain some of these letters, but TM can't find a, then TM terminates with **No**. Otherwise, TM finds the first instance of a, replaces a by a service symbol, say $*$, and adopts state q_a. Being in the state q_a, TM returns to the beginning of the input.

Stage 2. TM acts as described in Stage 1, except q_c is replaced in the description by q_a and a by b. If TM did not terminate, it ends up being in state q_b and observing the beginning of the word.

Stage 3. TM acts as in Stage 2, replacing in the description q_a by q_b and b by c. If TM does not terminate, it ends up in the state q_c, at the beginning of the input.

Now TM goes to Stage 1 and continues until it terminates (TM always terminates because the number of letters a, b, c in the input reduces on each stage, being replaced by $*$).

4. Here is a description of a TM recognizing palindromes in $\{a,\ b\}$.

The set of states of the Turing machine includes four states, q_a, q'_a, q_b and q'_b which are meant to "memorize" the letters a and b as well as the direction of movement of the working head. The machine reads the first letter, say a, of the input word w, erases it, adopts the state q_a and moves to the end of the word (being in the state q_a). If the last letter of the word does not match the state, i.e. is b, then the machine halts with **No**. Otherwise, the machine erases the last letter and moves one step to the left. Let the letter now observed be b. The machine adopts the state q'_b and moves to the beginning of the current word. Machine thus works comparing the pairs of letters in w which are at the same distance from the ends of w. The word is accepted if it consumes the input without rejection.