Exercise I

1) Formal definition:

An LA is a 6-tuple $(Q, \Sigma, \Gamma, \delta, q_0, F)$ where Q, Σ, Γ, and F are all finite sets and

1. Q is the set of states
2. Σ is the input alphabet
3. Γ is the list alphabet
4. $\delta : Q \times \Sigma \times \Gamma_\varepsilon \times \Gamma_\varepsilon \times \Gamma_\varepsilon \times \{0,1\} \rightarrow Q \times \Gamma_\varepsilon$ is the transition function
5. $q_0 \in Q$ is the start state
6. $F \in Q$ is the set of accept states.

2) Formal description of computation:

An LA $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ computes as follows. It accepts input w if w can be written as $w = w_1 w_2 \dots w_m$ where each $w_i \in \Sigma$ and a sequence of states $r_0, r_1, \dots, r_m \in Q$ and strings $l_0, l_1, \dots, l_m \in \Gamma^*$ exist that satisfy the following conditions:

1. $r_0 = q_0$ and $l_0 = \varepsilon$ (M starts out properly, in the start state and the list is empty)
2. For $i = 0, \dots, m-1$ we have $\delta(r_i, w_{i+1}, a_1, a_2, a_3, b) = (r_{i+1}, l_{i+1})$ for some $a_1, a_2, a_3 \in \Gamma_\varepsilon$ and $b \in \{0,1\}$. The condition we verify at every step is determined by b's value, as follows:
   i) $b = 1 \Rightarrow$ we check if $a_1 \in l_i$
   ii) $b = 0 \Rightarrow$ we check if $\cancel{a_1} \notin l_i$
3. $r_m \in F$ (r_m is an accept state)

The strings l_i represents the sequence of list contents.

Here is an example of an LA M that recognizes the language $L = \{ w \mid w$ contains an even number of characters 'b' $\}$. $M = (Q, \Sigma, \Gamma, \delta, q_0, F)$ where:

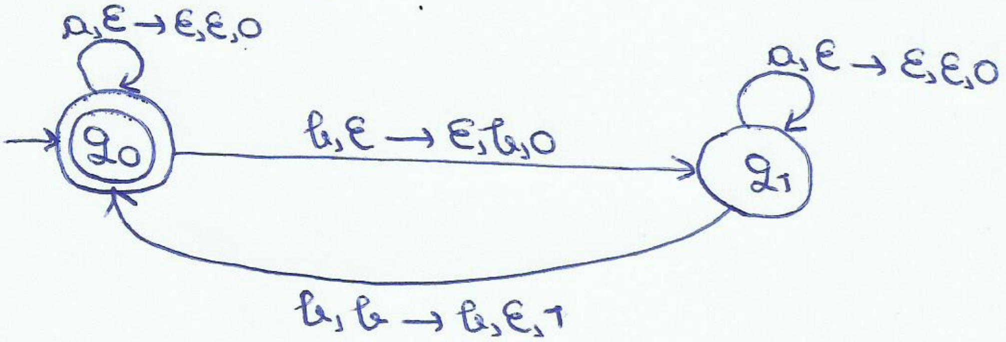
$Q = \{q_0, q_1\}$

$\Sigma = \{a, b\}$

$\Gamma = \{a, b, \varepsilon\}$

$F = \{q_0\}$

The diagram:



If the current input symbol is 'a', then we move to the next state without modifying the list. But, when the symbol is 'b' we verify if the list contains 'b'. If it does, then we remove the 'b' from the list. Otherwise, we add 'b' to the list.