Exercise 10

Task 1
The following MSO-formula was obtained via the technique from Büchi's Theorem (slide 154) from a regular language L. Which language is characterized by this formula?

$$
\begin{aligned}
&\exists X_1 \exists X_2 \exists X_3 (X_1 \cap X_2 = \emptyset \wedge X_1 \cap X_3 = \emptyset \wedge X_2 \cap X_3 = \emptyset \wedge \forall x (x \in X_1 \vee x \in X_2 \vee x \in X_3) \\
&\wedge \exists x (\forall y (x \leq y) \wedge ((P_a(x) \wedge x \in X_2) \vee (P_b(x) \wedge x \in X_2))) \\
&\wedge \exists x (\forall y (y \leq x) \wedge (x \in X_2)) \\
&\wedge \forall x \forall y (y = x + 1 \rightarrow (x \in X_1 \wedge P_a(y) \wedge y \in X_2) \\
&\qquad\qquad\qquad \vee (x \in X_1 \wedge P_b(y) \wedge y \in X_2) \\
&\qquad\qquad\qquad \vee (x \in X_2 \wedge P_a(y) \wedge y \in X_3) \\
&\qquad\qquad\qquad \vee (x \in X_2 \wedge P_b(y) \wedge y \in X_3) \\
&\qquad\qquad\qquad \vee (x \in X_3 \wedge P_a(y) \wedge y \in X_3) \\
&\qquad\qquad\qquad \vee (x \in X_3 \wedge P_a(y) \wedge y \in X_3)))
\end{aligned}
$$

Solution:
We find that the formula corresponds to the finite automaton $(\{1,2,3\}, \{a,b\}, \delta, 1, \{2\})$ with

$$
\begin{aligned}
\delta(1,a) &= 2 \\
\delta(1,b) &= 2 \\
\delta(2,a) &= 3 \\
\delta(2,b) &= 3 \\
\delta(3,a) &= 3 \\
\delta(3,b) &= 3.
\end{aligned}
$$

This automaton accepts the language $L = \{w \in \{a,b\}^+ \mid |w| \leq 1\}$.

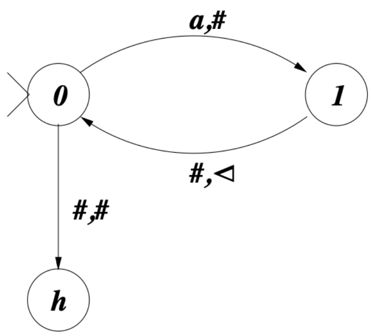
Task 2 (Turing Machines)
Given the alphabet $\Sigma = \{0,1\}$ and special symbol #, construct the following machines:

1. A Turing machine which decides the language consisting of palindromes over Σ.
2. A Turing machine that, once started, erases all the 1's from the head backwards, until it finds another #.

Solution:
Start by reading the first symbol and transition to state A if it's 0, or to state B if it's 1. Replace the symbol with a blank. Move the tape head to the right until reaching the end of the tape (marked by the first blank symbol). Then, move one symbol to the left. If this symbol is 0 and you're in state A, or if it's 1 and you're in state B, replace it with a blank and move all the way to the left until finding another blank symbol, then move one step to the right. If this condition is not met, it means the word is not a palindrome, and you stop and reject it. Repeat this process until you either reject the word or all symbols on the tape have been replaced with blanks, in which case you accept it. This procedure takes roughly $(n+1)+n+...+1 \approx O(n^2)$ moves.

1. Let us use Automata as description of this Turing Machine. As a remainder, automatas are graphs with labeled edges that represent the possible transitions and the vertices are the possible states. For example, a transition from the state p to the state q labeled as a, b means if the MT is in the state p and there is a letter a under the head, then the machine goes to the state q and performs the b action. The action of writing a letter $a \in \Sigma$ is simply a. The action of the head moving to the left or right is denoted as $\triangleleft$ or $\triangleright$ respectively. The action of writing the blank symbol is $\#$ and is also called *erase the cell.*

   With this description is easy to draw this machine as following:



Task 3
Let be $f : \Sigma_0^* \to \Sigma_1^*$, where $\# \notin \Sigma_0 \cup \Sigma_1$. We say that a Turing Machine $M = (K, \Sigma, \delta, s)$ computes f if

$$\forall w \in \Sigma_0^*, (s, \#w\ \underline{\#}) \vdash_M^* (h, \#f(w)\underline{\#})$$

The position of the Turing machine is stated by the underscore, i.e, the computation starts in the state s and the head is looking at the $\#$ after the input w. At the end of the computation the machine will be in a state h with the head looking at the first $\#$ after the output.

Given the alphabet $\Sigma = \{0, 1\}$ and special symbol $\#$, construct the following machine:

- A Turing Machine that calculates $f(w) = \bar{w}$, i.e., a machine that changes the 1's for 0's and viceversa in $w \in \{0,1\}^*$.

Solution:
The next automata represents the Turing Machine for our problem