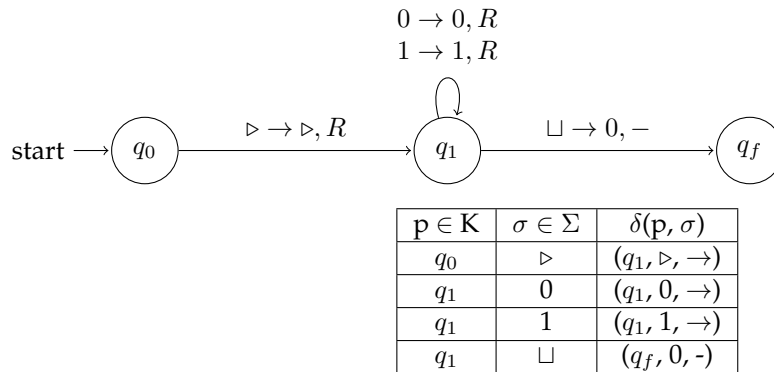


## Homework 1

### Exercise 1

**Solution:**



### Exercise 2

**Solution:**

According to the lecture notes, this is the definition about an ordinary Turing machine:

**Definition 3.1:** A Turing machine is a structure  $M = (K, \Sigma, \delta, s)$ , where  $K$  is a finite set of states,  $s \in K$  is the initial state, the alphabet  $\Sigma$  is a set of (tape) symbols, and where  $K$  and  $\Sigma$  are disjoint. We assume that  $\Sigma$  always contains the special symbols  $\sqcup$  and  $\triangleright$ , the blank and the first symbol. Finally,  $\delta$  is a transition function, where

$\delta: K \times \Sigma \rightarrow (K \cup \{h, \text{"yes"}, \text{"no"}\}) \times \Sigma \times \{\leftarrow, \rightarrow, -\}$ .

We assume that  $h$  (the halting state), "yes" (the accepting state), "no" (the rejecting state), and the cursor directions  $\leftarrow$  for "left",  $\rightarrow$  for "right" and  $-$  for "stay", are extra symbols not in  $K \cup \Sigma$ .

And if one were to admit TMs with countably many states that would look like this:

An infinite Turing machine is a quadruple  $M = (K; \Sigma; \delta; s)$ , where  $\Sigma$  and  $s$  are exactly as an ordinary Turing machine, but the set  $K$  must contain infinite states.  $\delta$  is a transition function that must reflect the complexities of infinite states. Intuitively,  $\delta$  decides the next state as before, but also contains infinite actions (steps). Nevertheless, the final states are unique, that is there should be only one  $h$  (the halting state), "yes" (the accepting state) and "no" (the rejecting state) in  $M$ .

So answering the question from the problem sheet, having countably infinitely many states doesn't open doors to solving problems that ordinary TM couldn't, but it might compute functions much faster since it has more states.