1 Construct an NFA for the language $(01)^+ + (010)^+$.

2 Construct an NFA that accepts all binary strings that have 1 as the third to last character; i.e., $x1ab$ for $a, b \in \{0, 1\}$ and $x \in \{0, 1\}^*$.

3 Given a DFA $M = (\Sigma, Q, \delta, s, A)$, construct an NFA N that accepts all prefixes of $L(M)$, i.e., $w \in L(N) \Leftrightarrow wx \in L(M)$ for some $x \in \Sigma^*$.

4 Given an DFA $M = (\Sigma, Q, \delta, s, A)$, construct an NFA N that accepts all suffixes of $L(M)$, i.e., $w \in L(N) \Leftrightarrow xw \in L(M)$ for some $x \in \Sigma^*$.

5 Given a DFA $M = (\Sigma, Q, \delta, s, A)$, construct an NFA N that accepts resverse of $L(M)$, i.e., $w \in L(N) \Leftrightarrow w^R \in L(M)$.

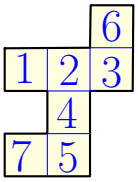
6 Given a DFA $M = (\Sigma, Q, \delta, s, A)$, construct an NFA N that accepts $insert1(L(M)) := \{x1y \mid xy \in L(M)\}$, i.e., strings in $L(M)$ with 1 inserted somewhere. For example, if $L(M) = \{\varepsilon, OOK!\}$, then $insert1(L(M)) = \{1, 1OOK!, O1OK!, OO1K!, OOK1!, OOK!1\}$.

Work on these later:

7 Given a DFA $M = (\Sigma, Q, \delta, s, A)$, construct an NFA N that accepts $delete1(L(M)) := \{xy \mid x1y \in L(M)\}$.

Intuitively, $delete1(L(M))$ is the set of all strings that can be obtained from strings in $L(M)$ by deleting exactly one 1. For example, if $L(M) = \{101101, 00, \varepsilon\}$, then $delete1(L(M)) = \{01101, 10101, 10110\}$.

8 Consider the following "Maze":



A robot starts at position 1 – where at every point in time it is allowed to move only to adjacent cells. The input is a sequence of commands V (move vertically) or H (move horizontally), where the robot is required to move if it gets such a command. If it is in location 2, and it gets a V command then it must move down to location 4. However, if it gets command H while being in location 2 then it can move either to location 1 or 3, as it chooses.

An input is ***invalid***, if the robot get stuck during the execution of this sequence of commands, for any sequence of choices it makes. For example, starting at position 1, the input HVH is invalid. (The robot was so badly designed, that if it gets stuck, it explodes and no longer exists.)

8.A. Starting at position 1, consider the (command) input HVV. Which location might the robot be in? (Same for $HVVV$ and $HVVVH$.)

8.B. Draw an NFA that accepts all valid inputs.

8.C. The robot *solves* the maze if it arrives (at any point in time) to position 7. Draw an NFA that accepts all inputs that are solutions to the maze.