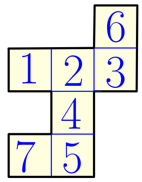
1 Consider the following "maze":



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 not valid. (The robot was so badly designed, that if it gets stuck, it explodes and no longer exists.)

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

Solution:

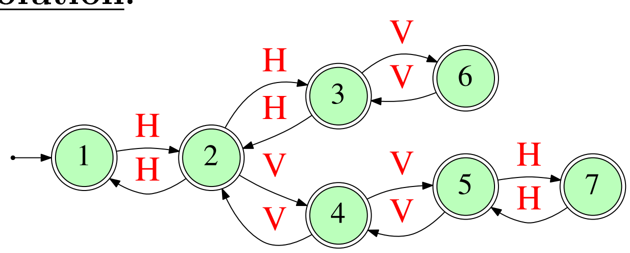
HVV: 2 or 5.

$HVVV$: 4.

$HVVVH$: This is an invalid input. The robot can not be in any valid location.

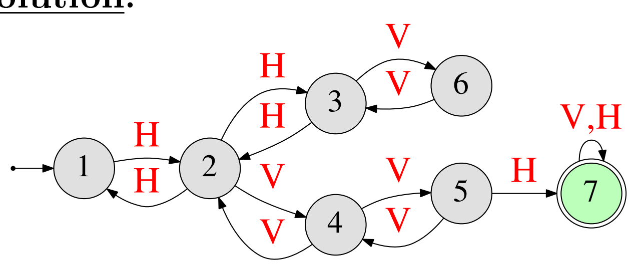
1 B. Draw an NFA that accepts all valid inputs.

Solution:



1 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.

Solution:



1 D. (Extra—not for discussion section.) Write a regular expression which is all inputs that are valid solutions to the maze.

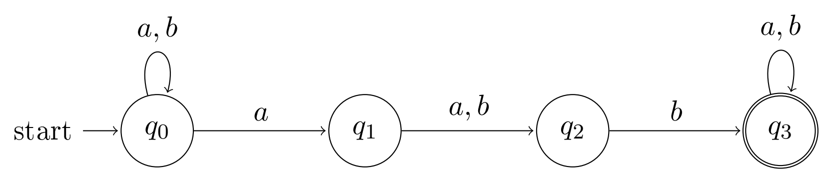
(See here for notes of how to solve such a question.)

2 Let $L = \{w \in \{a, b\}^* \mid a$ appears in some position i of w, and a b appears in position $i + 2\}$.

2 A. Create an NFA N for L with at most four states.

Solution:

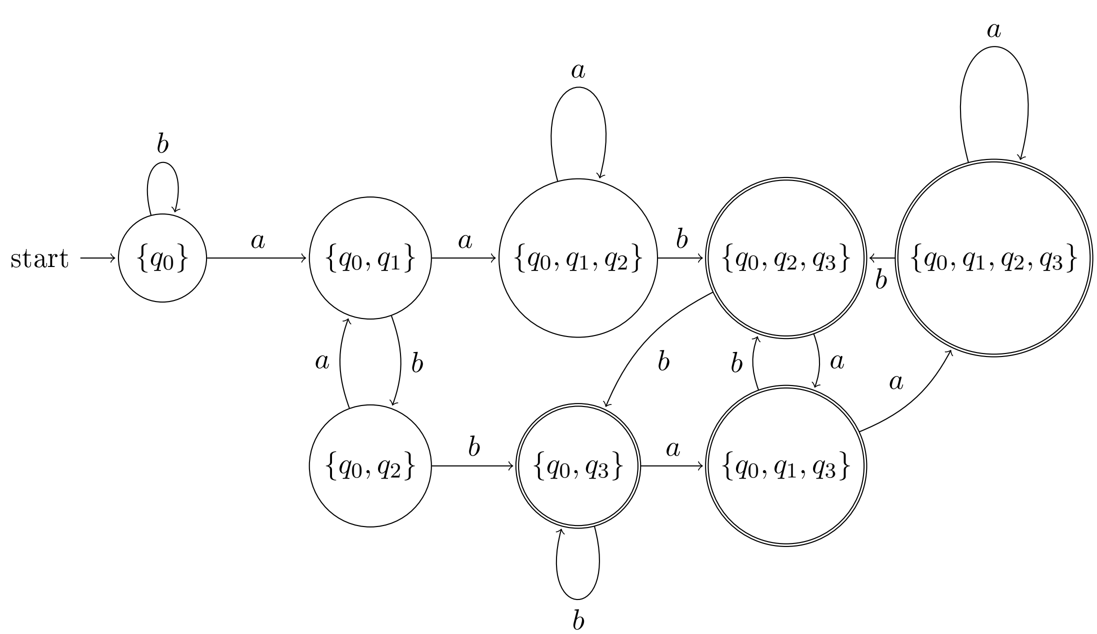
The following NFA N accepts the language. The machine starts at state q_0. On seeing the symbol a, the NFA has the choice of either staying at q_0 or to check if it is followed, 2 positions later, with a b.



2 B. Using the "power-set" construction, create a DFA M from N. Rather than writing down the sixteen states and trying to fill in the transitions, build the states as needed, because you won't end up with unreachable or otherwise superfluous states.

Solution:

Using the "power-set" construction, we obtain the following DFA M.



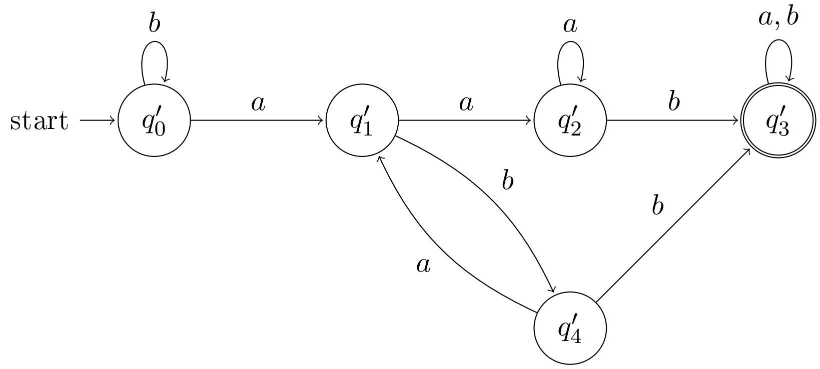
2 C. Now directly design a DFA M' for L with only five states, and explain the relationship between M and M'.

Solution:

The DFA M' is as follows. M' remembers the last two symbols seen so far.

- q'_0 is the start state. M'

- q_1' corresponds to having seen ba as the last two symbols (or just a if this is the first symbol).
- q_2' corresponds to having seen aa as the last two symbols.
- q_3' is the accepting state.
- q_4' corresponds to having seen ab as the last two symbols.



Note that if we contract all the accepting to states in M (from part (b)) to one state, then we obtain M'.