

CS 381 - A3

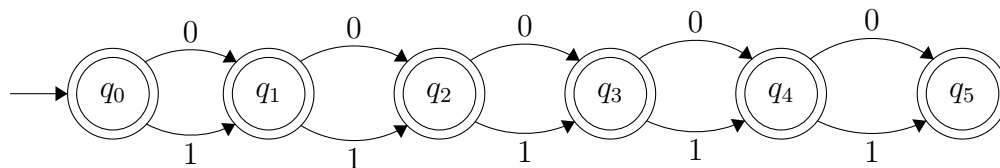
Martin Mueller

Due: Friday, February 28th, 2020

1. (10 points) Prove that there is a language $A \subseteq \{0, 1\}^*$, with both of the following properties:
 - (a) For all $x \in A$, $|x| \leq 5$.
 - (b) Every NFA that recognizes A has more than 5 states. Assume all NFAs here do not have λ -transitions.

Hint: Do not explicitly define an example language. Count the number of NFAs that have a certain number of states and then count the number of languages that are comprised of binary strings of certain lengths and see what happens. Your proof will be "non-constructive".

We start by explicitly defining the language A . Property (a) states that all elements in A must be 5 or fewer characters in length. Since A is a subset or equal to $\{0, 1\}^*$, we could classify A as the language containing all binary strings of lengths 0 to 5 or $\{\lambda, 0, 1, 00, 01, \dots, 11111\}$. In order to construct an NFA that recognizes this language, we would need to make it accept every binary string of lengths 0 to 5. This would require at least 6 accept states, one for each character of its input with transitions in one direction from itself to a state accepting more characters. The last accept state, accepting any binary string of length 5 would not have any transitions and therefore the NFA would not accept any string of a length greater than 5. An example NFA would be:



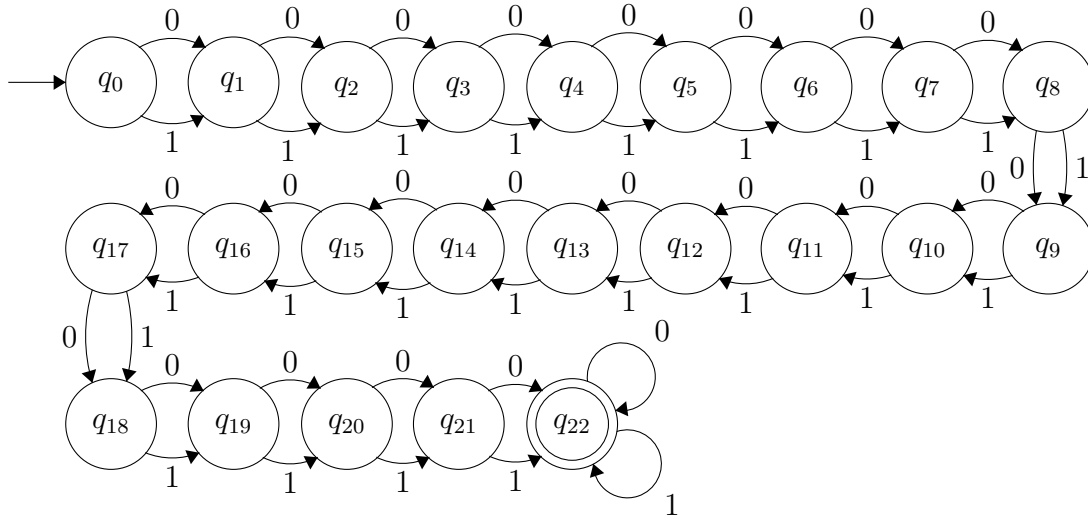
Since this NFA recognizes the language A , A is also regular.

2. (10 points) For this problem, consider the language consisting of the set of all binary strings of the form $xybz$, where $b \in \{0, 1\}$, $x, z \in \{0, 1\}^{10}$ and $y \in \{0, 1\}^*$. Show that this language is regular by constructing the smallest machine that you can think of that recognizes it. For fun (and a challenge), think about how many states will be required by any DFA that recognizes this language.

Give a digitally-drawn state diagram that is well-organized and the states have intuitive yet concise labels.

Give a concise and precise English description of how your machine works. If your machine has more than 10 states, then you should describe your reasoning that led you to construct your particular machine (I'm not saying you should try to construct a machine with 10 states or less).

Finally, give a formal description of your machine, contained in the file A3-2.nfa, that runs in the Automata-Simulator.



The machine starts at q_0 . From there, it must be given a string in $\{0, 1\}^{22}$ in order to reach the accept state. From there, any further input will just cause it to remain in the accept state. It's designed this way because $xybz$ is effectively:

$$\begin{aligned} xybz &= \{0, 1\}^{10} \{0, 1\} \{0, 1\}^* \{0, 1\} \{0, 1\}^{10} \\ &= \{0, 1\}^{11} \{0, 1\}^* \{0, 1\}^{11} \end{aligned}$$

As far as this DFA goes, this means that it needs to accept somewhere between 22 and infinite zeros or ones. It's pretty repetitive, so it's easy to prove this is the case.