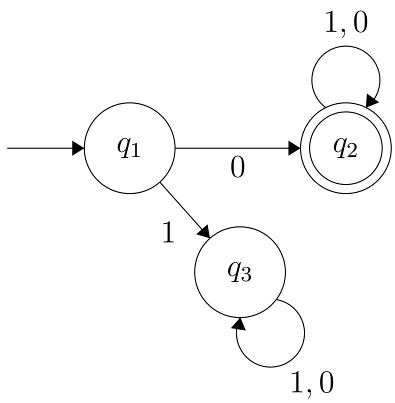
TOC PROBLEM SET-1

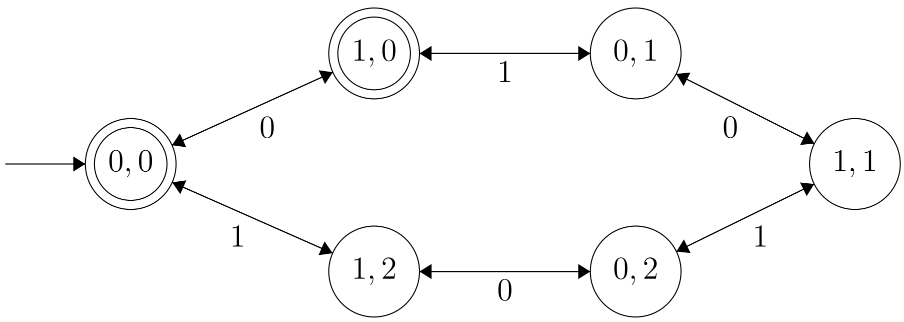
SIDDHANT CHAUDHARY
BMC201953

1. If possible, construct a DFA over the alphabet $\Sigma = \{0, 1\}$ that accepts a binary string w if and only if $(w)_{10}$ is divisible by:

(a) 2. This is an easy one. We need to accept only those words which end with a 0. A possible DFA for this is given below.



(b) 3. Here is the key observation: $2^k = 2(\text{mod } 3)$ if k is odd, and $2^k = 1(\text{mod } 3)$ if k is even. Note that we scan a word from right to left. So we maintain six states, which are labeled by an order pair of the form $(0, x)$ or $(1, x)$, where x is any residue mod 3, and 0 and 1 represent the number of bits scanned modulo 2. The transitions from one state to another is easy by the given congruences. So, the following DFA works:



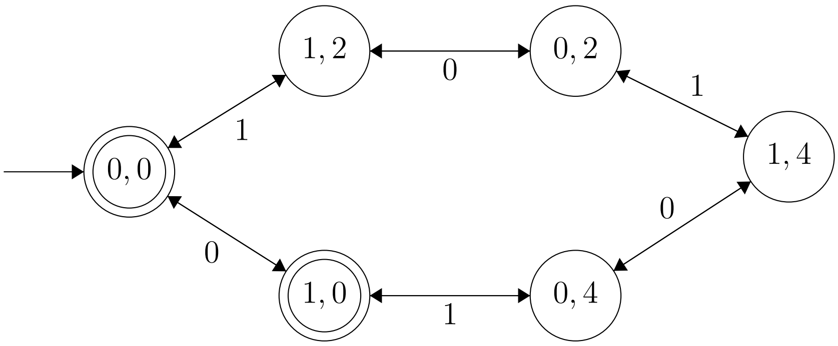
where the accepting states are only those which have a residue of 0 mod 3.

(c) 2 and 3. In this case, $(w)_{10}$ will be divisible by 6. We can repeat the exact same procedure as above, keeping in mind the following congruences:

$$2^k = \begin{cases} 1(\text{mod } 6) & \text{, if } k = 0 \\ 2(\text{mod } 6) & \text{, if } k = 1(\text{mod } 2)\text{and } k > 0 \\ 4(\text{mod } 6) & \text{, if } k = 0(\text{mod } 2)\text{and } k > 0 \end{cases}$$

Keeping these in mind, the DFA is the following:

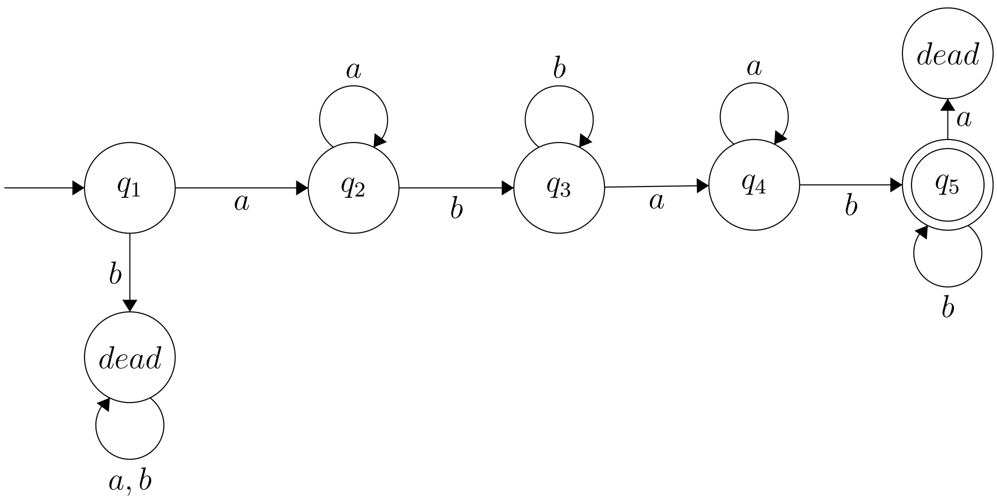
Date: August 2020.



I think in general, given a string in base 2, we can construct a DFA to check if $(w)_{10}$ is divisible by n by finding congruences for powers of 2.

2. Can you construct an NFA over $\Sigma = \{a, b\}$ that accepts a word w if and only if w contains exactly one ab and two ba's as in fixes (overlap is allowed)? Can you give a DFA that accepts the same language?

Solution: We give a DFA for this problem. First, note that the string will be scanned from right to left. We need two ba's and one ab. Either of these occurs if there is a letter switch. Observe that if the string begins on the right with a b, then it cannot have more ba's than ab's, and so in this case the desired output cannot be achieved. The only scenario in which we get two ba's and one ab is whe the string starts with an a on the right. We keep track on the number of ba's and ab's, and get the following DFA:



where $dead$ represents a dead state, i.e the possible configuration cannot be achieved.

3. Prove that for words u, v and an integer $e \geq 0$, $u(vu)^e = (uv)^e u$.

Solution: By induction on e. Base case for $e = 0$ is clear. Suppose it is true for $e - 1$. Then, we have

$$\begin{aligned} u(vu)^e &= u(vu)(vu)^{e-1} \\ &= (uv)u(vu)^{e-1} \\ &= (uv)(uv)^{e-1}u \\ &= (uv)^e u \end{aligned}$$

where induction was used in the second last step, and we used the fact that concatenation is associative.

4. Construct an automaton over the alphabet $\Sigma = \{a, b, c\}$ that accepts the language $L = \{w|\ |w|_a + |w|_b$ is odd$\}$.

Solution: Here, we only need to keep track of $|w|_a + |w_b|$ modulo 2. The following DFA is easy to construct: