P81

CMPS 130

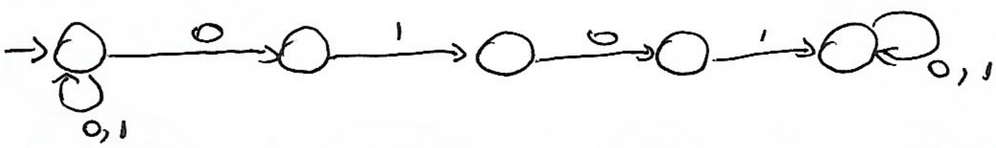
Computational Models
Homework Assignment 3
VLADOI MARIAN

Spring 2017

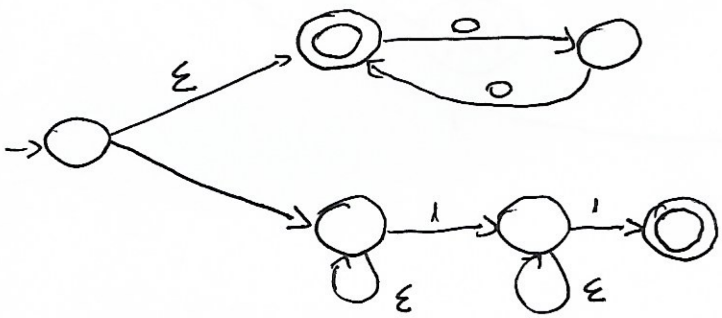
Exercise 1.7. page 84

Give the state diagrams of NFAs with the specific number of states recognizing each of the following languages. In all parts, the alphabet is $\{0,1\}$.

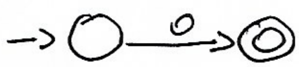
b) The language of exercise 1.6c with five states



c) The language of exercise 1.6l with six states

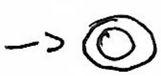


d) the language 0 with two states



e) The language $0^*1^*0^+$ with three states

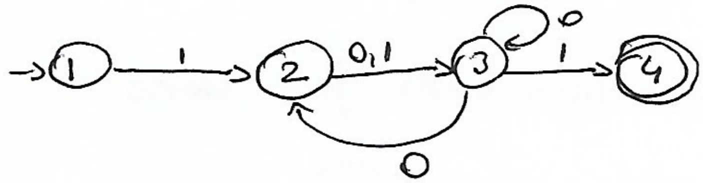
f) the language $\{\varepsilon\}$ with one state

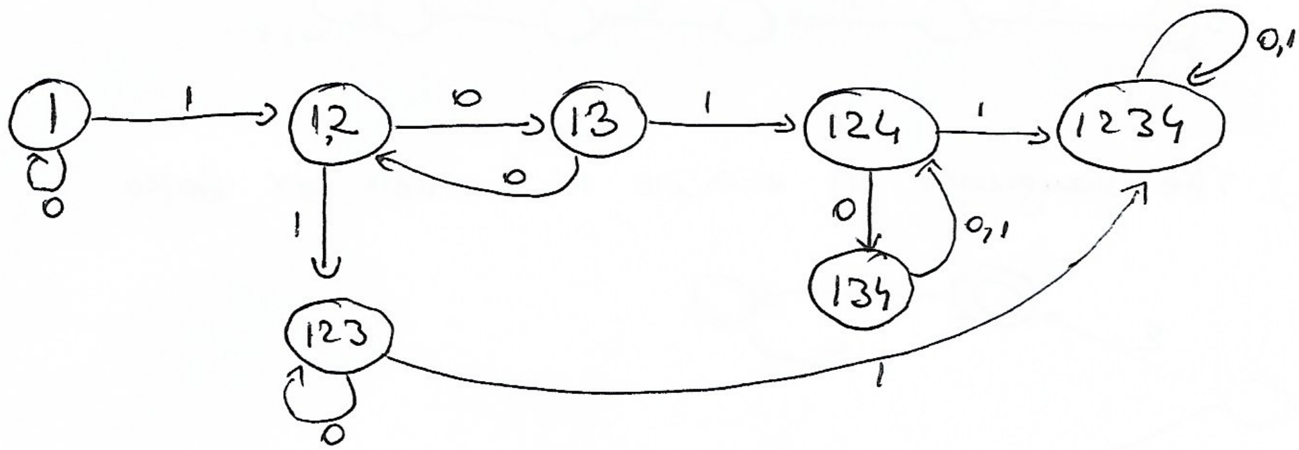


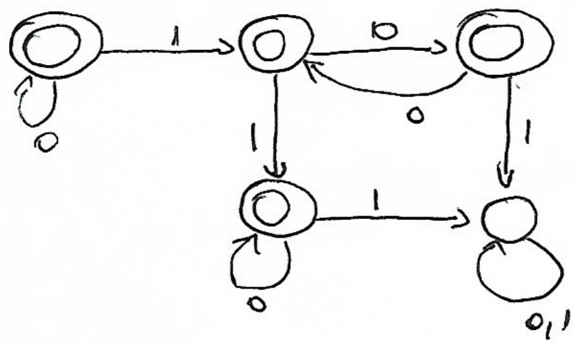
h) the language 0^* with one state

Exercise 1.13 page 85.

Let F be the language of all strings over {0,1}, that do not contain a pair of 1s, that are separated by an odd number of symbols. Give the state diagram of a DFA with five states that recognize F.



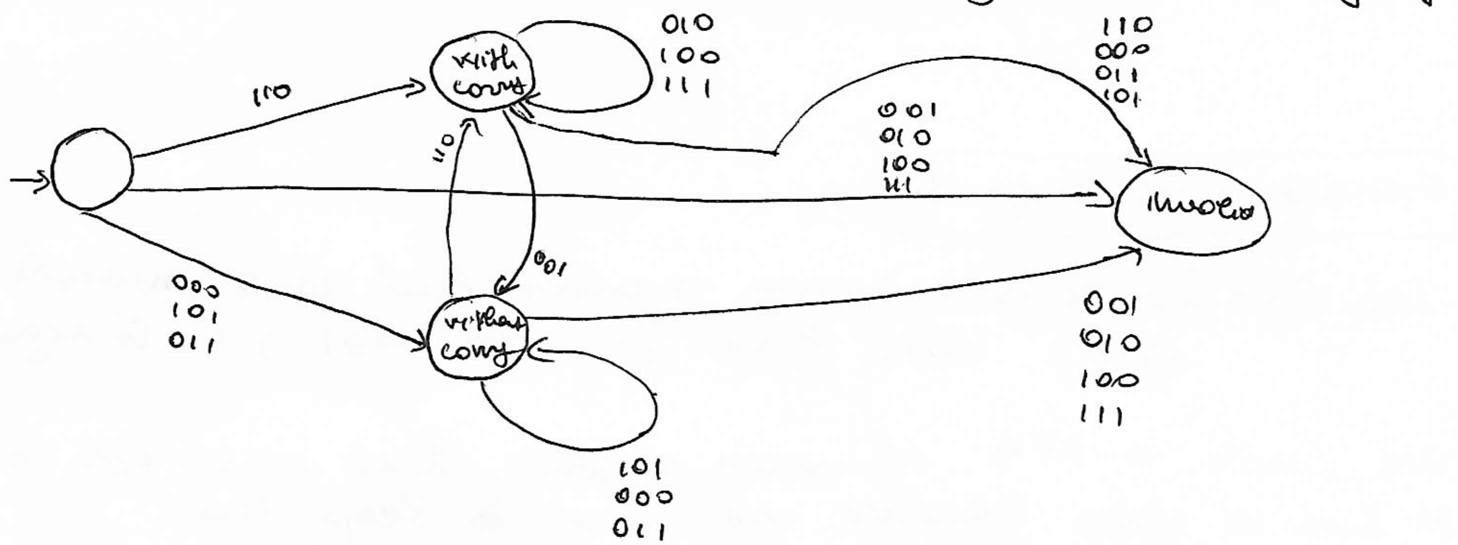




Exercise 1.32 page 88

Σ_3 contains all size of 3 columns of 0s and 1s. A string of symbols in Σ_3 gives three rows of 0s and 1s. Consider each row to be a binary number and let
$B = \{ w \in \Sigma_3^* \mid$ the bottom row of w is the sum of the top two rows $\}$. Show B is regular.

We will construct a NFA that recognize this language



Exercise 1.36 page 83.

Let $B_n = \{ a^k \mid k$ is a multiple of $n \}$. Show that for each $n \geq 1$, the language B_n is regular.

For each $n \geq 1$, we build a DFA with n states $q_0, q_1 \dots q_{n-1}$, to count the number of consecutive a's modulo n read so far.
Every time we scan an "a" the counter increments by 1 and jumps to the next state in M.
If and only if the machine stops at q_0, the string is accepted. this means that the length of the string is a multiple of n. Also, the string consists of all a's.

PS4

$Q = \{q_0, q_1, q_2, q_3 \ldots q_{n-1}\}$

$S = q_0$

$F = \{q_0\}$

$\Sigma = \{a\}$

$\delta(q_i, a) = q_j$ where $j = (i+1) \bmod n$.

Exercise 1.37 page 88

Let $C_n = \{x, x$ is a binary number that is a multiple of $n\}$. Show that for each $n \geq 1$, C_n is regular.

We create a DFA M with n states that recognize C_n.
M has n states because we have to keep track of the n possible remainders of the division.
The state correspond to the remainder 0 is the start state, and the only accept state.

We input the string to M from the most significant bit.
For each input M doubles the remainder.
Then M adds the input bits.
In the new state is the sum modulo n.
If the input string ends at the accept state, the binary number has no remainder on division by n.
The binary number $\in C_n$

$Q = \{q_0, q_1, q_2, \dots q_n\}$

$\Sigma = \{0, 1\}$

$S = q_0$

$F = \{q_0\}$

δ: $q_i \in Q$ and $b \in \{0,1\}$

$\delta(q_i, b) = q_j$, $\quad j = (2i + b) \bmod n$

Exercise 1.41. page 88

For languages A and B, let the perfect shuffle of A and B be the language
$\{w \mid w = a_1 b_1 \dots a_k b_k$, where $a_1 \dots a_k \in A$ and $b_1 \dots b_k \in B$, each $a_i, b_i \in \Sigma\}$.
Show this kind of regular languages is closed under perfect shuffle.

A is regular language => A is recognized by some finite automaton.

B is regular language => B is recognized by some finite automaton.

We have to prove that PerfectShuffle (A, B) is regular.

We have to construct a finite automaton M that recognize Perfect shuffle (A, B)

P86

$M_1 = (Q_1, \Sigma_1, \delta_1, s_1, F_1)$ M_1 recognize A
$M_2 = (Q_2, \Sigma_2, \delta_2, s_2, F_2)$ M_2 recognize B

M recognize PerfectShuffle (A, B).

$M = (Q, \Sigma, \delta, s, F)$

(1) $Q = Q_1 \times Q_2 \times \{1,2\}$

(2) $\Sigma = \Sigma_1 \cup \Sigma_2$

(3) $s = q_0 = (s_1, s_2, 1)$

(4) $F = F_1 \times F_2 \times \{2\}$

(5) $q_i \in Q_1$

$p_j \in Q_2$

$a \in \Sigma_1$

$\delta((q_i, p_j, 1), a) = (\delta_1(q_i, a), p_j, 2)$

$\delta((q_i, p_j, 1), a) =$

$=$ (empty set) $a \notin \Sigma_1$

$\delta((q_i, p_j, 2)\ a) = (q_i, \delta_2(p_j, a), 1)$

$\delta((q_i, p_j, 2), a) =$

$=$ (empty set) $a \notin \Sigma_2$