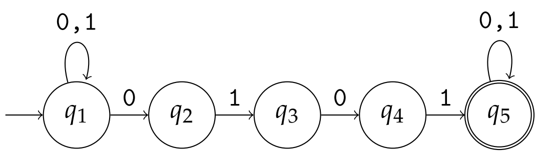
Homework 3

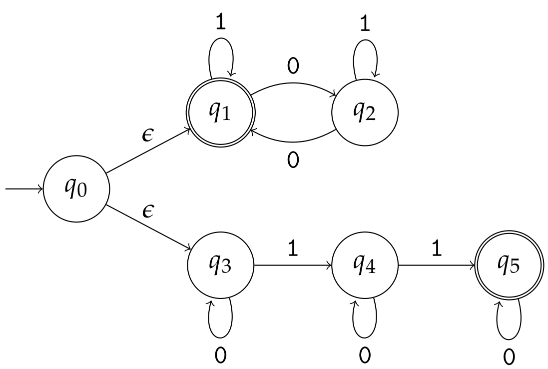
CMPS130 Computational Models, Spring 2015

1.7

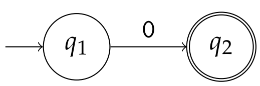
b.



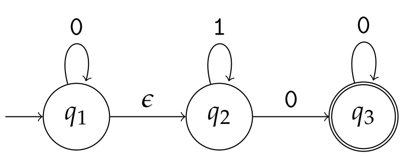
c.



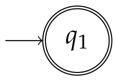
d.



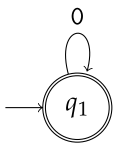
e.



g.

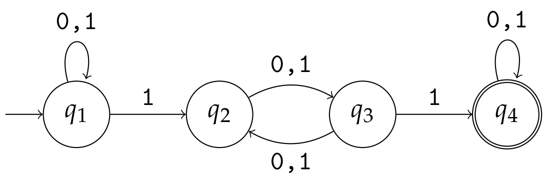


h.

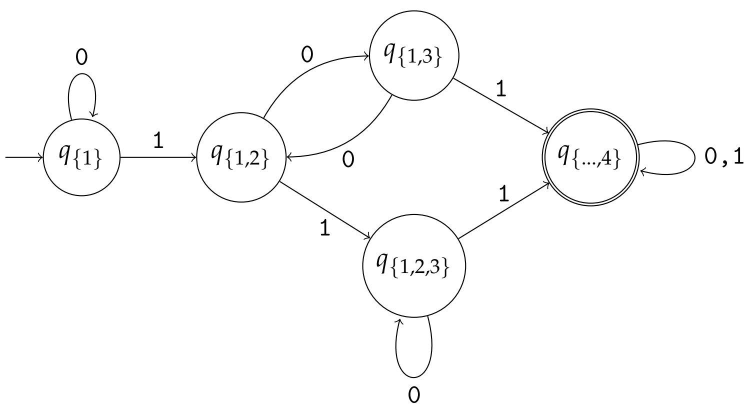


1.13

Starting with a 4-state NFA for the complement of F. This automata accepts any string does contain a pair of 1 with an odd number of symbols inbetween.



Now, a simple subset construction results in an equivalent DFA. If we merge the equivalent accepting states and then make every accepting state non-accepting and every non-accepting state accepting, we get a 5-state DFA for F.



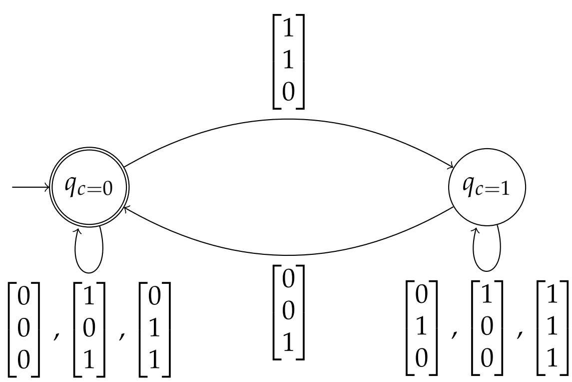
1.32

We can show that B is regular by constructing a NFA that accepts $B^{\mathcal{R}}$ and then use the result from 1.31 which implies that B is also regular.

Given a symbol $w = \begin{bmatrix} a \\ b \\ s \end{bmatrix}$, we can distinguish two different cases.

- If we don't have a carry ($c = 0$), then we only need to check that $a + b = s$, so acceptable combinations are $\{ \begin{bmatrix} 0 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 1 \end{bmatrix} \}$. If $a = 1$ and $b = 1$, then s would 10 in binary, so we check that $s = 0$ in the combination $\begin{bmatrix} 1 \\ 1 \\ 0 \end{bmatrix}$ and remember a carry $c = 1$ for processing the next digit in the number.
- If we do have a carry ($c = 1$), then we need to check that $a + b + 1 = s$, so acceptable combinations are $\{ \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix}, \begin{bmatrix} 0 \\ 1 \\ 0 \end{bmatrix}, \begin{bmatrix} 1 \\ 1 \\ 1 \end{bmatrix} \}$. All of these result in a carry except for the case with $a = 0$ and $b = 0$.

This results in the following NFA which sums up binary numbers by starting at least significant bit (thereby accepting $B^{\mathcal{R}}$ rather than B) and consuming elements one at a time, while storing the current carry with two states $q_{c=0}$ and $q_{c=1}$.



This NFA accepts a sequence of binaries numbers with the third row being the sum of the first two numbers. This language is therefore regular. By reversing all sequences such that the most significant bit comes first, the language will still be regular.

1.36

Let $B_n = \{\mathtt{a}^k \mid \text{where } k \text{ is a multiple of } n\}$. For each $n \geq 1$, we can show that B_n is regular by constructing an NFA $N = (Q, \Sigma, \delta, q_0, F)$ that accepts the language.

Let $Q = \{q_0\} \cup \{q_i \mid 1 \leq i < n\}$ and $F = \{q_0\}$. Here, q_0 denotes the states in which the current count of a's is a multiple of n. The remaining states q_i denote i more a's than the last multiple of n.

The transition function δ uses states to count a's. If there are exactly n a's since the last multiple of n then this is another multiple of n, so the transition goes back to q_0.

$$\delta(q_i, \mathtt{a}) = \begin{cases} q_{i+1} & \text{if } \ i+1 < n \\ q_0 & \text{otherwise} \end{cases}$$

Proof.

- $w \in B_n \quad \Rightarrow \quad \hat{\delta}(q_0, w) \in F$.
  Given a word $w = a^k$, the NFA will do k transitions. Each n'th transition will go back to q_0, so if k is a multiple of n, the final state will be $q_0 \in F$ and therefore accepting.
- $\hat{\delta}(q_0, w) \in F \quad \Rightarrow \quad w \in B_n$.
  Given a word $|w| = k$ such that the NFA will do k transitions before reaching an accepting state. q_0 is the only accepting state and it can be reached in two different ways.
  - If q_0 was reached simply by stopping in the initial state without doing any transitions, then $w = \epsilon$ and $k = 0$, which is a multiple of n, so $w \in B_n$.
  - If q_0 was reached by going through n transitions from the last time it was in q_0, then $w = w'a^n$. From a simple induction it follows, that w' is also either ϵ or a multiple, so w will be $w^{c \cdot n}$, a multiple of n and therefore $w \in B_n$.

1.37

Let $C_n = \{x \mid x \text{ is a binary number that is a multiple of } n\}$. For each $n \geq 1$, we can show that C_n is regular by constructing an NFA $N = (Q, \Sigma, \delta, q_0, F)$ that accepts the language.

Let $Q = \{q_0\} \cup \{q_i \mid 1 \leq i < n\}$ and $F = \{q_0\}$. Here, q_0 denotes the states in which the binary interpretation of the current prefix of w is an exact multiple of n. The remaining states q_i denote $w = c \cdot n + i$, i.e. the binary interpretation of w is i more than the next smaller multiple of n.

The transition function δ goes to the correct state depending on the next symbol in the input. The key insight is that we can start with the most significant bit first and then an additional 0 would simply multiply the number by two and an additional 1 at the end would multiply it by two and increase is by one.

$$\begin{array}{ccccc} \mathtt{101} & \rightarrow & \cdot\mathtt{0} & \rightarrow & \mathtt{1010} \\ 5 & \rightarrow & *2 & \rightarrow & 10 \end{array} \qquad\qquad \begin{array}{ccccc} \mathtt{101} & \rightarrow & \cdot\mathtt{1} & \rightarrow & \mathtt{1011} \\ 5 & \rightarrow & *2+1 & \rightarrow & 11 \end{array}$$

$$\delta(q_i, \mathtt{0}) = q_{(i \cdot 2 \bmod n)} \qquad\qquad \delta(q_i, \mathtt{1}) = q_{(i \cdot 2 + 1 \bmod n)}$$

Proof.

- $w \in C_n \;\;\Rightarrow\;\; \hat{\delta}(q_0, w) \in F$.
  Given a word w which is the binary representation of a number x which a multiple of n. This means w has the form $w = w_k w_{...} w_2 w_1 w_0$ and $x = w_k \cdot 2^k + ... + w_2 \cdot 2^2 + w_1 \cdot 2 + w_0$. With factorization, it becomes $x = ((w_k) \cdot 2 + w_{k-1}) \cdot 2 + w_{k-2}....$ Let x_i be the binary representation of the w prefix of length i. It can be shown by induction that after i transitions, the NFA will be in state q_j such that $j = x_i \bmod n$. Because x is a multiple of n, $x_k \bmod n = 0$ and therefore the NFA will be in state q_0 which is accepting.

- $\hat{\delta}(q_0, w) \in F \;\;\Rightarrow\;\; w \in C_n$.
  Given a word $w = w_k ... w_1 w_0$ such that the NFA reaches the accepting state q_0, it can be shown by induction that the NFA will be in state q_j only if $j = x_i \bmod n$.
  - If q_0 was reached simply by stopping in the initial state without doing any transitions, then $w = \epsilon$ and $k = 0$, which is a multiple of n, so $w \in B_n$.
  - If q_0 was reached from another state q_i, then either $w_0 = 0$ and therefore the $i \cdot 2 \bmod n = 0$ or $w_0 = 1$ and $i \cdot 2 + 1 \bmod n = 0$. In each case, $x_k \bmod n = 0$, so x is a multiple of n and w, its binary representation, will be in the language $w \in B_n$.

1.41

The perfect shuffle PS of two regular languages A and B is also regular.

$$PS(A, B) = \{a_1 b_1 a_2 b_2 ... a_k b_k \mid a_1 a_2 .. a_k \in A \;\wedge\; b_1 b_2 ... b_k \in B\}$$

For all regular languages A and B, we can show that $PS(A, B)$ is also regular by constructing a DFA that accepts the language $PS(A, B)$. Since A and B are regular languages, we can assume there are DFAs $D_A = (Q_A, \Sigma_A, \delta_A, q_{A,0}, F_A)$ and $D_B = (Q_B, \Sigma_B, \delta_B, q_{B,0}, F_B)$.

$$N = (Q, \Sigma, \delta, q_0, F) \qquad \Sigma = \Sigma_A \cup \Sigma_B$$
$$Q = Q_A \times Q_B \times \{A, B\} \cup \{q_{err}\} \qquad \delta(\langle q_A, q_B, A\rangle, s) = \langle \delta_A(q_A, s), q_B, B\rangle$$
$$q_0 = \langle q_{A,0}, q_{B,0}, A\rangle \qquad \delta(\langle q_A, q_B, B\rangle, s) = \langle q_A, \delta_B(q_B, s), A\rangle$$
$$F = \{\langle q_a, q_b, A\rangle \mid q_a \in F_A \;\wedge\; q_b \in F_B\} \qquad \delta(\langle q_A, q_B, M\rangle, s) = q_{err} \text{ if } s \notin \Sigma_M$$

The DFA D keeps track of the DFA state in D_A, the DFA state in D_B and whether the next symbol is expected to be for D_A or D_B. Here, $< q_a, q_b, M >$ denotes the tuple of state $q_a \in Q_A$, $q_b \in Q_B$ and $M \in \{A, B\}$. The language described by the DFA D is regular but we still need to show that it actually accepts the perfect shuffle $PS(A, B)$ for each regular language A and B.

Proof.

- $w \in PS(A, B) \;\;\Rightarrow\;\; \hat{\delta}(q_0, w) \in F$.
  Given a word $w = a_1 b_1 a_2 b_2 ... a_k b_k$, the DFA will do $2k$ transitions. Each odd transition updates q_A based on δ_A and each even transition updates q_B based on δ_B. Because $a_1 a_2 .. a_k \in A$, it follows that $\hat{\delta}(q_{A,0}, a_1 a_2 .. a_k) \in F_A$ and, $\hat{\delta}(q_{B,0}, b_1 b_2 .. b_k) \in F_B$, therefore $\hat{\delta}(q_0, w) \in F$.

- $\hat{\delta}(q_0, w) \in F \;\;\Rightarrow\;\; w \in PS(A, B)$.
  Given a word $w = a_1 b_1 a_2 b_2 ... a_k b_k$ such that the DFA reaches an accepting state, all odd symbols in w will be accepted by D_A, so $a_1 a_2 ... a_k \in A$ and similarly $b_1 b_2 ... b_k \in B$, so $a_1 b_1 a_2 b_2 ... a_k b_k \in PS(A, B)$.