1. To formalize and generalize the DFA M_i, let $0 \leq i \leq k+1$ So that $M_i = (Q_i, \Sigma_i, \delta_i, q_0, F_i)$.

   M_i accepts (w| w is a string of elements such that the sum of all elements mod i = 0. or in other words a multiple of i).

   Where

   $Q_i = \{q_0, q_1, ..., q_{i-1}\}$

   $\Sigma_i = \{0, 1, ..., i, < RESET >\}$

   $\delta_i\ (q_j, a) \Rightarrow q_k : a \in \Sigma, k = (j + a)\ mod\ i,\ \ , j =$ the current running sum of each element read into M_i since the first element or since the most recent ¡RESET¿, whichever occurred most recently.

   $F_i = \{q_0\}$

2. a)

   the DFA $D_1, = (Q_1, \Sigma, \delta_1, q_1, F_1)$.

   D_1 accepts (w| w starts with an 'a').

   Where

   $Q_1 = \{q_1, q_2, q_3\}$

   $\Sigma = \{a, b\}$

   $\delta_1 =\{\ (q_1, a) \Rightarrow q_2$

   $(q_1, b) \Rightarrow q_3$

   $(q_2, \{a, b\}) \Rightarrow q_2$

   $(q_3, \{a, b\}) \Rightarrow q_3$
   }

   $F_1 = \{q_2\}$

   the DFA $D_2, = (Q_2, \Sigma, \delta_2, q_1'', F_2)$.

   D_2 accepts (w| w has at most 2 b's).

   Where

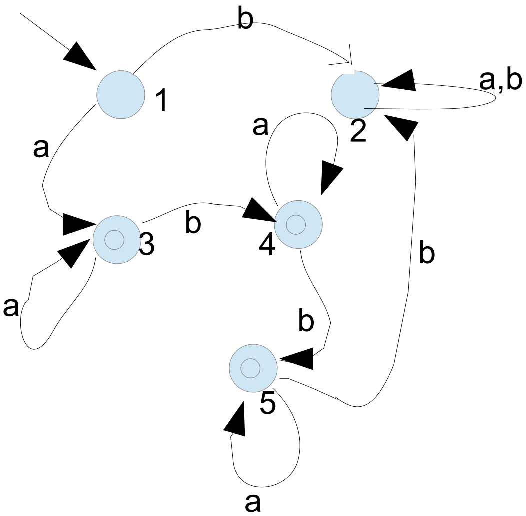
   $Q_2 = \{q_1'', q_2'', q_3'', q_4''\}$

   $\Sigma = \{a, b\}$

   $\delta_2 =\{\ (q_1'', a) \Rightarrow q_1''$

   $(q_1'', b) \Rightarrow q_2''$

DFA- D3



$(q''_2, a) \Rightarrow q''_2$

$(q''_2, b) \Rightarrow q''_3$

$(q''_3, a) \Rightarrow q''_3$

$(q''_3, b) \Rightarrow q''_4$

$(q''_4, a) \Rightarrow q''_4$

$(q''_4, b) \Rightarrow q''_4$ }

$F_2 = \{q''_1, q''_2, q''_3\}$ }

Our goal is to create a new machine D_3 which serves as a machine which recognizes every language recognizable by D_1 and D_2. in other words $\text{L}(D_3) =$ the $\text{L}(D_1) \cap \text{L}(D_2)$

we will define D_3 as follows

$D_3, = (Q_3, \Sigma, \delta_3, q'''_1, F_3)$.

D_3 accepts (w| w begins with an a and has at most two b's).

Where

$Q_3 = \{Q_1 * Q_2\}$

$\Sigma = \{a, b\}$

δ_3 :Let $q_1, q''_1 \in Q_3 : a \in \Sigma$

$= \{ \ ((q_1, q''_1), a) \Rightarrow \delta_1(q_1, a), \delta_2(q''_1, a)$

$F_3 = \{(F_1 * Q_2) \cup (Q_1 * F_2)\}$

$q'''_1 = (q_1, q''_1)$;

---

b)

the DFA $D_4, = (Q_4, \Sigma, \delta_4, q_1, F_4)$.

D_4 accepts (w| w has odd length).

Where

$Q_4 = \{q_1, q_2\}$

$\Sigma = \{a, b\}$

$\delta_4 = \{ \ (q''_1, (a, b)) \Rightarrow q''_2$

$(q_2'', (a,b)) \Rightarrow q_1''$

$F = \{q_2\}$ }

the DFA $D_5, = (Q_5, \Sigma, \delta_5, q_1'', F_5)$.

D_5 accepts (w| w has an even number of a's).

Where

$Q_5 = \{q_1'', q_2''\}$

$\Sigma = \{a, b\}$

$\delta_5 =\{\ (q_1'', a) \Rightarrow q_2''$

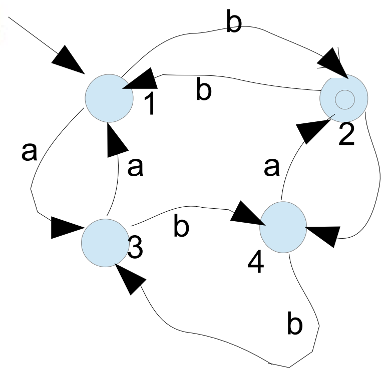
$(q_1'', b)) \Rightarrow q_1''$

$(q_2'', a) \Rightarrow q_1''$

$(q_2'', b)) \Rightarrow q_2''$

$F_5 = \{q_1\}$ }

DFA-- D_7



Our goal is to create a new machine D_7 which serves as a machine which recognizes every language recognizable by D_4 and D_5. in other words L(D_7) = the L$(D_4) \cap$ L(D_5)

$D_7 = (Q_7, \Sigma, \delta_7, q_1''', F_7)$.

D_7 accepts (w| w has odd length and an even number of 'a's or).

Where $Q_7 = \{Q_4 * Q_5\}$

$\Sigma = \{a, b\}$

δ_7 :Let $q_1, q_1'' \in Q_7 : a \in \Sigma$

$= \{\ ((q_1, q_1''), a) \Rightarrow \delta_4(q_1, a), \delta_5(q_1'', a)$

$F_7 = \{(F_4 * Q_5) \cup (Q_4 * F_5)\}$

$q_1''' = (q_1, q_1'')$;

3. a)

Let a simple DFM be M_2 which accepts any string in $a^* \cup b^*$

we define M_2 as follows:

$M_2 = (Q_2, \Sigma, \delta_2, q_1, F_2)$.

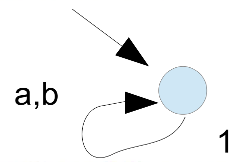
$Q_2 = \{q_1\}$

$\Sigma = \{a, b\}$

$\delta_2 =\{\ (q_1, (a,b)) \Rightarrow q_1$

$F_2 = \{q_1\}$ }

We can now, simply switch every state that accepts a word to a an unaccepting state. In this case the start state is the only in the machine, and the complement of M_2 is the machine which accepts no words, not even the empty word.



---

b)

Let a simple DFM be M_3 which accepts any string with exactly one (a or b) (we are interpreting the exercise to mean if an element is an 'a' or a 'b' it counts as an (a or b) and if another (a or b) is read it counts as two (a or b)'s)

we define M_3 as follows:

$M_3 = (Q_3, \Sigma, \delta_3, q_1, F_3)$.

$Q_3 = \{q_1, q_2, q_3\}$

$\Sigma = \{a, b\}$

δ_3 ={ $(q_1, (a, b)) \Rightarrow q_2$

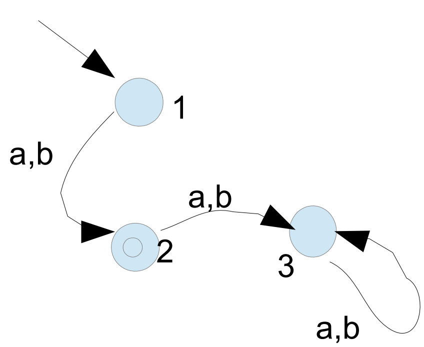
$(q_2, (a, b)) \Rightarrow q_3$

$(q_3, (a, b)) \Rightarrow q_3$

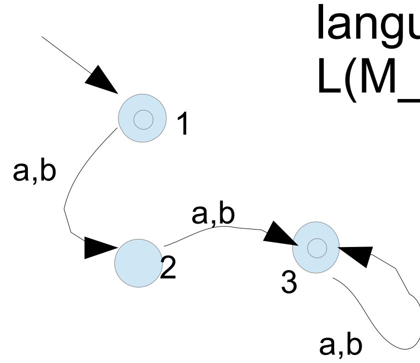
$F_3 = \{q_2\}$ }

We can now, simply switch every state that accepts a word to that of a unaccepting state, and every nonacceptance state to an accepting state. In this way we will construct the machine which accepts the complement of the language accepted by M_3

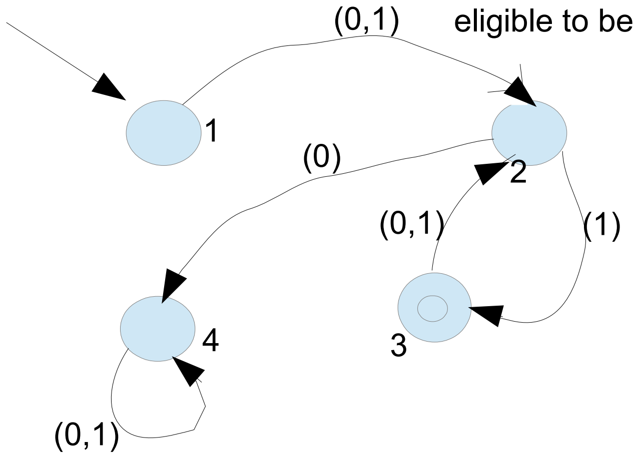
DFA-- M_3



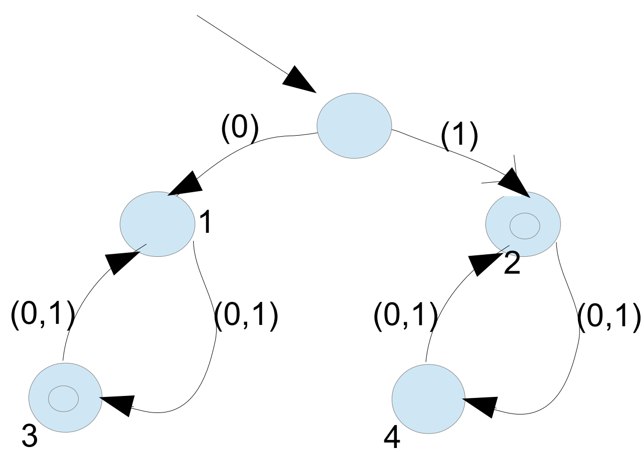
DFA-- accepts the cmplmnt language to L(M_3)



4.a) let the first element read in from the start state represent the first position which is odd. That is to say: the start state is the 0th position and can not be a one, therefore at least two elements are required for a word to be eligible to be accepted.



4.b)



4.c)